Intel® 3100 Chipset

Datasheet

June 2007

Order Number: 313458-004US

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. See http://www.intel.com/products/processor_number for details.

The MICH may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at http://www.intel.com.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino logo, Core Inside, Dialogic, FlashFile, i960, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, IPLink, Itanium, Itanium Inside, MCS, MMX, Oplus, OverDrive, PDCharm, Pentium, Pentium Inside, skoool, Sound Mark, The Journey Inside, VTune, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2007, Intel Corporation. All Rights Reserved.

Contents

1.0 Introduction 48
1.1 System Architecture 49
1.2 Supported Microprocessors 49
1.3 Supported Memory Devices 50
1.4 PCI Express* 51
1.5 Supported Debug and Management Interfaces 51
1.6 Supported IMCH Integrated Features 52
1.6.1 EDMA Controller 52
1.6.2 Integrated Memory Init/Test Engine 52
1.6.3 Coherent Memory Write Buffer 52
1.6.4 RASUM Features 53
1.7 IMCH Feature List 54
1.7.1 FSB Interface 54
1.7.2 Memory Interface 54
1.7.3 PCI Express* Interface in IMCH 55
1.7.4 EDMA Controller 56
1.7.5 Coherent Memory Write Buffer 57
1.7.6 Integrated Memory Scrub Engine 57
1.7.7 Hardware Memory Initialization Engine 58
1.7.8 System Management Functions 58
1.7.9 RASUM 58
1.8 IICH Feature List 59
1.8.1 PCI Express* Interface in IICH 59
1.8.2 Low-Pincount (LPC) Interface and Firmware Hub (FWH) Interface 59
1.8.3 Integrated Serial ATA (SATA) Host Controllers 59
1.8.4 USB 60
1.8.5 Interrupt Controller 60
1.8.6 Power Management Logic 60
1.8.7 DMA Controller 60
1.8.8 Timers Based on 82C54 60
1.8.9 High Precision Event Timers (HPET) 60
1.8.10 Real Time Clock with 256-byte Battery-backed CMOS RAM 61
1.8.11 System TCO Reduction Circuits 61
1.8.12 SMBus 61
1.8.13 Watchdog Timer 61
1.8.14 PCI Bus Interface 62
1.8.15 Serial Port 62
1.8.16 GPIO 62

2.0 Configuration Register Descriptions 63
2.1 General Register Information 63
2.1.1 Register Attributes 63
2.1.2 Register Nomenclature 63
2.2 IMCH Registers Summary 63
2.2.1 IMCH Configuration Register Maps 64
2.2.2 IMCH Configuration Register Summaries 69
2.2.3 IMCH Memory Mapped Registers 75
2.3 IICH Registers Summary 79
2.3.1 IICH Configuration Register Maps 79
2.3.2 IICH Memory Mapped Registers 91

3.0 Enhanced Direct Memory Access Controller (EDMA) 110

3.1 Overview 110
3.1.1 Features 111
3.1.2 Logical Block Diagram 112
3.2 Channel Programming Interface 113
3.3 Chaining Operation 114
3.3.1 Chain Descriptor Definition 114
3.3.2 EDMA Chain Descriptor in Memory 115
3.3.3 Chain Descriptor Usage 115
3.3.4 Scatter/Gather Transfer 117
3.3.5 Appending to a Descriptor Chain 117
3.3.6 Splicing a Descriptor Chain into a Linked List 118
3.4 Transfer Types 119
3.4.1 Local Memory to Local Memory 119
3.4.2 Local Memory to I/O Subsystem Memory 119
3.4.3 I/O Memory to Local Memory 120
3.4.4 I/O Memory to I/O Memory 120
3.5 Addressing 120
3.5.1 Address Coherence 120
3.5.2 Addressing Modes 121
3.5.3 PCI Express A Port Traffic Class 132
3.6 Channel Data Queuing 132
3.7 Error Conditions 132
3.7.1 Controller Interface Error 134
3.7.2 Memory Interface Error 134
3.7.3 I/O Interface Error 135
3.8 Channel Arbitration 135
3.8.1 Normal Arbitration Scheme 135
3.8.2 Prioritized Arbitration Scheme 135
3.9 Configuration 136
3.9.1 Power Up/Default Status 137
3.9.2 Channel-Specific Register Definitions 137
3.10 Interrupts 141
3.10.1 Interrupt Routing Mechanisms 142
3.10.2 Message Signaled Interrupt (MSI) 143
3.10.3 Interrupt Ordering 144
3.11 Initiating an EDMA Transfer 145
3.11.1 Setup and Initiation 145
3.11.2 Suspend Function 146
3.11.3 Stop Function 146
3.11.4 EDMA Process Flow 147

4.0 Power Requirements and Interface Signals 149
4.1 Power Supply Requirements 150
4.2 Power Wells 150
4.3 Power Planes 153
4.4 Signal Information 154
4.4.1 Signaling Technology 154
4.5 Pin Straps 175

5.0 System Address Map 176
5.1 Memory Map 176
5.1.1 System Memory Spaces 177
5.1.2 VGA and MDA Memory Spaces 178
5.1.3 PAM Memory Spaces 180
5.1.4 ISA Hole Memory Space 184
5.1.5 TSEG SMM Memory Space 185

5.1.6 PCI Express* Enhanced Configuration Aperture 185
5.1.7 IOAPIC Memory Space 186
5.1.8 FSB Interrupt Memory Space 186
5.1.9 High SMM Memory Space 187
5.1.10 PCI Device Memory (MMIO) 187
5.2 IMCH Responses to EDMA Transactions 188
5.2.1 Fixed Address Spaces (EDMA) 188
5.2.2 Relocatable Address Spaces (EDMA) 189
5.3 I/O Address Space 190
5.3.1 Configuration Window 190
5.3.2 VGA and MDA Regions 190
5.4 Main Memory Addressing 191
5.5 System Management Mode (SMM) Space 191
5.5.1 SMM Addressing Ranges 192
5.6 Memory Reclaim Background 193
5.6.1 Memory Remapping Algorithm 193
5.7 IICH Register and Memory Mappings 194
5.7.1 I/O Map 194
5.7.2 Memory Map 197
5.7.3 Boot-Block Update Scheme 198

6.0 Platform Configuration 200
6.1 RASUM Features - SMBus and TAP Access 200
6.2 Platform Configuration Structure Conceptual Overview 200
6.2.1 IMCH PCI Devices 201
6.2.2 IICH PCI Devices 202
6.3 Routing Configuration Accesses 204
6.3.1 Standard PCI Bus Configuration Mechanism 204
6.3.2 PCI Bus #0 Configuration Mechanism 204
6.3.3 Primary PCI and Downstream Configuration Mechanism 205
6.3.4 IMCH PCI Express Bus Configuration Mechanism 206
6.3.5 IMCH Configuration Cycle Flow Chart 207
6.4 IMCH Register Introduction 207
6.5 IMCH Sticky Registers 208
6.6 IMCH I/O Mapped Registers 209
6.6.1 CONFIG_ADDRESS - Configuration Address Register 209
6.6.2 CONFIG_DATA - Configuration Data Register 209
6.7 IMCH Memory Mapped Registers 210
6.8 PCI Express* Enhanced Configuration Mechanisms 210
6.8.1 PCI Express* Configuration Transaction Header 210
6.8.2 Enhanced Configuration Hardware Implications 211
6.8.3 Enhanced Configuration Memory Address Map 211
6.8.4 Enhanced Configuration FSB Address Format 211

7.0 RAS Features and Exception Handling 213
7.1 RAS Features 213
7.1.1 Data Protection 213
7.1.2 DRAM Data Integrity 214
7.1.3 PCI Express* Data Integrity 216
7.1.4 Test/Support Major Busses 217
7.2 Exception Handling 217
7.2.1 FERR/NERR Global Register Scheme 217
7.3 Error Conditions Signaled 223

8.0 Platform Management (IMCH) 226
8.1 Integrated SMBus Interface 226

8.2 SMBus Target Architecture 226
8.2.1 High Level Operation 226
8.2.2 Suggested SMBus Usage Models 234
8.3 PCI Express* Hot-Plug Capability 235
8.3.1 Architectural Support for Hot-Plug 235
8.3.2 Initialization For Hot-Plug 242
8.3.3 Port Configuration Implications of Hot-Plug 243
8.4 Platform Power Management Support 243
8.4.1 Supported System Power States 244
8.4.2 FSB Interface Power Management 244
8.4.3 DDR2 Interface Power Management 245
8.4.4 PCI Express* Interface Power Management 245
8.4.5 Device and Slot Power Limits 248
8.4.6 PME Support 248
8.4.7 BIOS Support for PCI Express* PM Messaging 250

9.0 System Clocking 252
9.1 IMCH Clocking 252
9.1.1 PCI Express* Clocking 252
9.1.2 DDR2 Geared Clocking 253
9.1.3 PCI Express* Port A (PEA) Platform Clocking Illustration 253
9.1.4 Spread-Spectrum Clocking Limitations 254
9.2 IICH System Clocking 255

10.0 System Reset 256
10.1 Reset 256
10.1.1 IMCH Reset Types 256
10.1.2 Reset Sequencing 258
10.2 IICH Reset 260
10.2.1 IICH System Reset References 260
10.2.2 NSI Reset Sequence 261

11.0 Packaging 262
11.1 Package Specifications and Quadrant Layout 262
11.2 Interface Trace Length Compensation 279
11.2.1 Other Interface Signal Package Trace Length Data 285

12.0 Supported DRAM Technology 286
12.1 Memory Interface 286
12.2 Memory Interface Performance Optimizations 286
12.2.1 DDR2 Overlapped Command Scheduling 286
12.2.2 Aggressive Page-Closed Policy with Look-Ahead 286
12.3 Memory Address Translation Tables 287
12.4 DIMM Topologies 287
12.4.1 DDR2-400 DIMM Slot Populations 287
12.4.2 DQ/DQS Mapping 290
12.5 Interface Signaling Voltage 290
12.6 Clock Gearing Ratios 291
12.7 Supported DRAM Timings 291
12.7.1 On-Die Termination (ODT) 291

13.0 IMCH Registers 293
13.1 Device 0, Function 0: IMCH Registers 293
13.1.1 Register Details 295
13.2 Device 0, Function 1: DRAM Controller Error Reporting Registers 344
13.2.1 Register Details 346
13.3 Device 1, Function 0: EDMA Registers 394

13.3.1 Register Details 395
13.4 Device 2, Function 0: PCI Express* Port A Standard and Enhanced Registers 413
13.4.1 Register Details 415
13.5 Device 3, Function 0: PCI Express* Port A1 Standard and Enhanced Registers 474
13.5.1 Register Details 474
13.6 Device 8, Function 0: Extended Configuration Registers 476
13.6.1 Register Details 476
13.7 Memory Mapped I/O for DDR2 Registers 485
13.7.1 Register Details 486
13.8 Memory Mapped I/O for EDMA Registers 511
13.8.1 Register Details 512

14.0 Bridging and Configuration (IICH) 531
14.1 Configuration Registers (Memory Space) 531
14.1.1 VC Configuration Registers 533
14.1.2 Root Complex Topology Capability Structure Registers 536
14.1.3 Internal Link Configuration Registers 541
14.1.4 I/O Data Bus Configuration Registers 543
14.1.5 TCO Configuration 544
14.1.6 Interrupt Configuration Registers 544
14.1.7 General Configuration Registers 549

15.0 Device 30, Function 0: PCI to PCI Bridge 555
15.1 Overview 555
15.2 Configuration Registers 555
15.2.1 Register Summary Table 555
15.2.2 PCI Header 556
15.2.3 Bridge Proprietary Configuration 567
15.2.4 PCI Bridge Vendor Capability 572
15.2.5 Manufacturing Information 573
15.3 PCI 32/33 Bus Interface 573
15.4 PCI Bridge as an Initiator 573
15.4.1 Memory Reads and Writes 574
15.4.2 I/O Reads and Writes 574
15.4.3 Configuration Reads and Writes 574
15.4.4 Locked Cycles 574
15.4.5 Target/Master Aborts 574
15.4.6 Secondary Master Latency Timer 574
15.4.7 Dual Address Cycle (DAC) 574
15.4.8 Memory and I/O Decode to PCI 575
15.5 PCI Bridge as a Target 576
15.5.1 Memory Writes 576
15.5.2 Terminology 576
15.6 Parity Error Detection and Generation 577
15.7 PCI Reset 578

16.0 Device 31, Function 0: LPC Interface 579
16.1 Overview 579
16.2 Configuration Registers (LPC I/F – D31, F0) 579
16.2.1 PCI Configuration Registers 580
16.2.2 ACPI/GPIO Configuration Registers 584
16.2.3 Interrupt Configuration Registers 586
16.2.4 LPC I/O Configuration Registers 587
16.2.5 Power Management Configuration Registers 590
16.2.6 FWH Configuration Registers 591
16.2.7 Root Complex Register Block Configuration Register 595

16.2.8 Manufacturing Information Register 595
16.3 Interface 596
16.3.1 Overview 596
16.3.2 Cycle Types 596
16.3.3 Aborting a Cycle 597
16.3.4 Memory Cycle Notes 597
16.3.5 I/O Cycle Notes 598
16.3.6 DMA Cycle Notes 598
16.3.7 Bus Master Cycle Notes 598
16.3.8 FWH Cycle Notes 598
16.3.9 LPC PD# Protocol 598
16.3.10 Cycle Posting Policies 599
16.3.11 Configuration 599
16.3.12 SERR# Generation 599

17.0 LPC DMA 601
17.1 Overview 601
17.2 DMA I/O Registers 602
17.2.1 Register Descriptions 603
17.3 DMA Channel Arbitration 609
17.4 Special Cases in Address/Count 609
17.4.1 Address Overrun/Underrun 609
17.4.2 16-Bit Channels 610
17.4.3 Autoinitialize 610
17.4.4 Software Commands 610
17.5 Theory of Operation for LPC DMA 610
17.5.1 Asserting DMA Requests 611
17.5.2 Abandoning DMA Requests 611
17.5.3 General Flow of DMA Transfers 612
17.5.4 Terminal Count 612
17.5.5 Verify Mode 612
17.5.6 DMA Request Deassertion 613
17.5.7 SYNC Field/LDRQ# Rules 613

18.0 8254 Timers 615
18.1 Overview 615
18.2 8254 Timer Register Summary Table 615
18.2.1 Counter 0, System Timer 615
18.2.2 Counter 1, Refresh Request Signal 616
18.2.3 Counter 2, Speaker Tone 616
18.2.4 Counter Operating Modes 616
18.3 Timer I/O Registers (LPC I/F – D31, F0) 616
18.3.1 Register Details 616
18.4 Timer Programming 618
18.5 Reading from the Interval Timer 618
18.5.1 Simple Read 618
18.5.2 Counter Latch Command 619
18.5.3 Read Back Command 619

19.0 Interrupts 622
19.1 Overview 622
19.2 8259 Interrupt Controllers (PIC) 624
19.2.1 Overview 624
19.2.2 I/O Registers 625
19.2.3 Interrupt Handling 632
19.2.4 Initialization Command Words (ICW) 633

19.2.5 Operation Command Words (OCW) 634
19.2.6 Modes of Operation 634
19.2.7 End of Interrupt (EOI) operations 636
19.2.8 Masking Interrupts 636
19.2.9 Steering of PCI Interrupts 636
19.3 Advanced Interrupt Controller: APIC 637
19.3.1 Interrupt Handling 637
19.3.2 PCI/PCI Express* Message-Based Interrupts 637
19.3.3 Memory Registers 639
19.3.4 Supporting External Interrupt Controllers 643
19.4 PCI Interrupts via /PCI Express* 643
19.5 Serial Interrupt 643
19.5.1 Overview 643
19.5.2 Start Frame 644
19.5.3 Data Frames 644
19.5.4 Stop Frame 645
19.5.5 Serial Interrupts Not Supported via SERIRQ 645
19.5.6 Special Notes on IRQ14 and IRQ15 645
19.5.7 Data Frame Format 645

20.0 Processor Interface 647
20.1 I/O Registers Associated with Processor Interface 648
20.1.1 Register Descriptions 649
20.2 Processor Interface Signals 651
20.2.1 A20M# (Mask A20) 651
20.2.2 INIT# (Initialization) 652
20.2.3 FERR#/IGNNE# (Coprocessor Error/Ignore Numeric Error) 652
20.2.4 NMI (Non-Maskable Interrupt) 653
20.2.5 INTR# (Interrupt Signals) 653
20.2.6 STPCLK# and CPUSLP# (Stop Clock Request and Processor Sleep Signals) .. 653
20.2.7 Enhanced Intel SpeedStep Technology (EIST) Signals 653
20.2.8 DPSLP# (Deeper Sleep) 653

21.0 Real Time Clock (LPC I/F – D31: F0) 654
21.1 Overview 654
21.2 RTC I/O Registers 654
21.3 Indexed Registers 655
21.3.1 Register Descriptions 656
21.4 Update Cycles 659
21.5 Interrupts 660
21.6 Lockable RAM Ranges 660
21.7 Century Rollover 660
21.8 Month and Year Alarms 660

22.0 Power Management 661
22.1 The Features 661
22.2 System Power States 661
22.3 Power Planes 663
22.3.1 System Power Planes 663
22.3.2 Power Planes 663
22.4 IMCH-IICH Messages 664
22.5 Power Management Registers 664
22.5.1 Power Management Registers in PCI Device 31, Function 0 665
22.5.2 APM I/O Decode 670
22.5.3 General I/O Decode Ranges for Power Management 671
22.6 SMI#/SCI Generation 690

22.7 Dynamic Processor Clock Control 693
22.7.1 Overview 693
22.7.2 Transition Rules Among S0/Cx and Sx States 694
22.7.3 S0/C0, S0/C2, Entry/Exit Timings and Sequences 694
22.8 Sleep States 697
22.8.1 Sleep State Overview 697
22.8.2 Initiating Sleep States 697
22.8.3 Exiting Sleep States 698
22.8.4 Sx-G3-Sx, Handling Power Failures 699
22.9 Processor Thermal Management 700
22.9.1 THRM# Signal for SMI# or SCI 700
22.9.2 Processor Initiated Passive Cooling 700
22.9.3 Force THRM# Throttle Software Bit 701
22.9.4 Active Cooling 701
22.10 Event Input Signals, Messages and Their Usage 701
22.10.1 PWRBTN# – Power Button 701
22.10.2 RI# – Ring Indicate Signal 702
22.10.3 PME# – PCI Power Management Event 703
22.10.4 SYS_RESET# Button 703
22.10.5 Processor Thermal Trip 703
22.10.6 SATA SCI 704
22.10.7 PCI Express WAKE# Signal and PME Event Message 704
22.10.8 PCI Express Hot Plug 704
22.11 Alternate (ALT) Access Mode 704
22.11.1 Write Only Registers with Read Paths in Alternate Access Mode 705
22.11.2 PIC Reserved Bits 707
22.11.3 Read-Only Registers with Write Paths in ALT Access Mode 707
22.12 System Power Supplies, Planes, and Signals 707
22.12.1 Power Plane Control with SLP_S3#, SLP_S4# and SLP_S5# 707
22.12.2 SLP_S4# and Suspend-To-RAM Sequencing 708
22.12.3 PWROK Signal 708
22.12.4 CPUPWRGD Signal 709
22.12.5 Controlling Leakage and Power Consumption During Low-Power States 709
22.12.6 VRMPWROK 709
22.13 Legacy Power Management Theory of Operation 709
22.13.1 Overview 709
22.13.2 APM Power Management 709

23.0 System Management 711
23.1 Overview 711
23.2 TCO Register Map 712
23.2.1 TCO PCI Configuration Registers 712
23.2.2 TCO I/O-Mapped Registers 712
23.3 TCO Signal Usage 719
23.3.1 INTRUDER# Signal 719
23.3.2 Pin Straps 720
23.3.3 SMLINK Signals 720
23.4 TCO Theory of Operation 720
23.4.1 Overview 720
23.4.2 Detecting a DOA CPU or System 720
23.4.3 Handling an Operating System Lockup 721
23.4.4 Handling a CPU or Other Hardware Lockup 721
23.4.5 Handling an Intruder 722
23.4.6 Handling a Potentially Failing Power Supply 722
23.4.7 Handling an ECC Error or Other Memory Error 722

23.4.8 SMM to Operating System and Operating System to SMM Calls 722
23.4.9 Detecting an Improper FWH Programming 723
23.4.10 IRQ1 and IRQ12 for Legacy Elimination 723
23.5 Event Reporting via SMLink/SMBus 724
23.5.1 Overview 724
23.5.2 Message Format 728
23.5.3 Connecting an External LAN Controller 728

24.0 Device 31, Function 0: General Purpose I/O 730
24.1 Overview 730
24.2 General Purpose I/O Registers (D31, F0) 730
24.2.1 GPIO Register Address Map 731
24.2.2 Register Descriptions 732
24.3 Additional GPIO Theory of Operation 737
24.3.1 Power Wells 737
24.3.2 SMI# and SCI Routing 737
24.3.3 Triggering 738
24.4 GPIO Summary Table 738

25.0 Device 29, Functions 0, 1: USB(1.1) Controllers 740
25.1 USB1 Configuration Registers 740
25.1.1 Register Details 741
25.2 USB I/O Registers 750
25.2.1 Register Details 750
25.3 Data Transfers to/from Main Memory 759
25.4 Data Structures in Main memory 759
25.5 Data Transfers To/From Main Memory 760
25.5.1 Executing the Schedule 761
25.5.2 Processing Transfer Descriptors 761
25.5.3 Command Register, Status Register, and TD Status Bit Interaction 762
25.5.4 Transfer Queuing 762
25.6 USB Buffer Management 766
25.7 Data Encoding and Bit Stuffing 766
25.8 Bus Protocol 767
25.8.1 Bit Ordering 767
25.8.2 SYNC Field 767
25.8.3 Packet Field Formats 767
25.8.4 Address Fields 767
25.8.5 Frame Number Field 767
25.8.6 Data Field 767
25.8.7 Cyclic Redundancy Check (CRC) 768
25.9 Packet Formats 768
25.10 .USB Interrupts 768
25.10.1 Overview 768
25.10.2 Transaction-Based Interrupts 768
25.10.3 Non-Transaction Based Interrupts 770
25.11 USB Power Management 771
25.12 USB Legacy Keyboard Operation 771

26.0 Device 31, Function 3: SMBus Controller Functional Description 774
26.1 Overview 774
26.1.1 Host Controller 774
26.1.2 Slave Interface 775
26.2 SMBus Controller Configuration Registers 775
26.2.1 Register Descriptions 776
26.3 I/O Registers 781

26.3.1 Register Descriptions 782
26.4 Host Controller 791
26.4.1 Overview 791
26.4.2 Command Protocols 792
26.4.3 I2C Behavior 801
26.4.4 Heartbeat for Use with External LAN 802
26.5 Bus Arbitration 802
26.6 Bus Timings 802
26.6.1 Clock Stretching 803
26.6.2 Bus Time Out (Intel® 3100 Chipset as SMB Master) 803
26.7 Interrupts/SMI# 803
26.8 CRC Generation and Checking 804
26.9 Slave Interface I/O Space 804
26.9.1 Register Details 805
26.10 Slave Interface Behavioral Description 808
26.10.1 Format of Slave Write Cycle 808
26.10.2 Format of Read Command 809
26.10.3 Format of the Host Notify Command 811

27.0 High Precision Event Timers (HPET) 813
27.1 Overview 813
27.2 Registers 813
27.2.1 Register Details 814
27.3 Theory Of Operation 821
27.3.1 Timer Accuracy Rules 821
27.3.2 Interrupt Mapping 822
27.3.3 Periodic vs. Non-Periodic Modes 822
27.3.4 Enabling the Timers 824
27.3.5 Interrupt Levels 824
27.3.6 Handling Interrupts 824
27.3.7 Issues Related to 64-bit Timers with 32-bit Processors 825
27.3.8 Unloading Device Driver Issues 825

28.0 Device 29, Function 7: USB 2.0 Host Controller 826
28.1 Overview 826
28.2 USB 2.0 Configuration Registers 826
28.2.1 Register Details 828
28.3 Memory-Mapped I/O Registers 846
28.3.1 Host Controller Capability Registers 846
28.3.2 Host Controller Operational Registers 849
28.4 EHC Initialization 863
28.4.1 Power On 863
28.4.2 Driver Initialization 864
28.4.3 EHC Resets 864
28.5 Data Structures in Main Memory 864
28.6 USB 2.0 Enhanced Host Controller DMA 864
28.6.1 Periodic List Execution 865
28.6.2 Asynchronous List Execution 867
28.7 Data Encoding and Bit Stuffing 870
28.8 Packet Formats 870
28.9 USB 2.0 Interrupts and Error Conditions 870
28.9.1 Aborts on USB 2.0-Initiated Memory Reads 870
28.9.2 Host Interface Parity Errors 871
28.10 USB 2.0 Power Management 873
28.10.1 Pause Feature 873
28.10.2 Suspend Feature 874

28.10.3 ACPI Device States 875
28.10.4 ACPI System States 875
28.11 Interaction with Classic Host Controllers 875
28.11.1 Port-Routing Logic 876
28.11.2 Device Connects 877
28.11.3 Device Disconnects 877
28.11.4 Effect of Resets on Port-Routing Logic 878
28.12 USB 2.0 Legacy Keyboard Operation 878
28.13 USB 2.0 Based Debug Port 878
28.13.1 USB 2.0 Based Debug Port Overview 879
28.13.2 Debug Port Registers 879
28.13.3 USB 2.0 Based Debug Port Theory of Operation 883

29.0 Device 31, Function 2: SATA Host Controller 888
29.1 Overview 888
29.1.1 PCI Header 888
29.1.2 Additional SFF-8038i Configuration Registers 898
29.1.3 Register Details 899
29.1.4 PCI Power Management Capabilities 903
29.1.5 Message Signaled Interrupt Capability 905
29.1.6 Additional Configuration Registers 907
29.2 I/O Registers 917
29.2.1 Primary Devices 917
29.2.2 Secondary Devices 919
29.2.3 Indirect AHCI Addressing Index/Data registers 920
29.3 Memory Registers 920
29.3.1 Generic Host Controller 921
29.3.2 Vendor Specific Registers 925
29.3.3 Port DMA Registers 925
29.3.4 Port Interface Registers (one set per port) 934
29.4 Overview 943
29.5 Legacy Operation 943
29.5.1 Transfer Examples 944
29.5.2 Error Handling 946
29.5.3 Hot Plug Operation 948
29.5.4 48-Bit (“Large”) LBA Operation Requirements 949
29.5.5 Power Management Operation 949
29.5.6 SATA Interrupts 952
29.5.7 Staggered Spin-up 953
29.5.8 Hardware/Software Operation for Detecting SATA Device Presence 953
29.5.9 SATA LED 954
29.5.10 Staggered Spin-up Support 954
29.6 AHCI Operation 954
29.6.1 System Memory Structures 954
29.6.2 Hot Plug Operation 955
29.6.3 Power Management Operation 957

30.0 Device 28, Functions 0, 1, 2, 3: PCI Express* Root Ports B 958
30.1 Overview 958
30.2 Configuration Registers 958
30.2.1 PCI Header 958
30.2.2 Root Port Capability Structure 968
30.2.3 Message Signaled Interrupt Capability 979
30.2.4 PCI Bridge Vendor Capability 981
30.2.5 PCI Power Management Capability 981
30.2.6 Port Configuration 983

30.2.7 Manufacturing Information 986
30.2.8 VC Configuration 986
30.2.9 Advanced Error Reporting Configuration 990
30.2.10 Root Complex Topology Capability Structure 998
30.3 PCI Express* Operation 1000
30.3.1 Interrupt Generation 1000
30.3.2 Power Management 1000
30.3.3 SERR# Generation 1002
30.3.4 Hot Plug 1006

31.0 Serial I/O Unit and Watchdog Timer 1010
31.1 Overview 1010
31.2 Features 1010
31.3 Signal Description 1011
31.4 Functional Description 1012
31.4.1 Host Processor Interface (LPC) 1012
31.5 LPC Interface 1012
31.5.1 LPC Cycles 1012
31.5.2 Policy 1013
31.5.3 LPC Transfers 1013
31.6 Logical Devices 4 and 5: Serial Ports (UART1 and UART2) 1014
31.6.1 UART Feature List 1015
31.6.2 Signal Descriptions 1015
31.6.3 UART Operational Description 1016
31.6.4 Internal Register Descriptions 1017
31.6.5 FIFO Operation 1030
31.7 Logical Device 6: Watchdog Timer 1031
31.7.1 Overview 1031
31.7.2 I/O Registers 1032
31.7.3 Theory Of Operation 1040
31.8 Serial IRQ 1042
31.8.1 Timing Diagrams For SIW_SERIRQ Cycle 1042
31.9 Configuration 1045
31.9.1 Configuration Port Address 1045
31.9.2 Primary Configuration Address Decoder 1045
31.9.3 SIW Configuration Register Summary 1046

32.0 Electrical Characteristics 1053
32.1 Absolute Maximum Ratings 1053
32.2 Power Characteristics 1053
32.3 I/O Interface Signal Groupings 1055
32.4 DC Characteristics 1061

Figures

1 Intel® 3100 Chipset Block Diagram 48
2 Device 0, Function 0 Registers 64
3 Device 0, Function 1 Registers 65
4 Device 1, Function 0 EDMA Registers 66
5 Devices 2 and 3, Function 0 PEA0 and 1 Port Standard and Enhanced Registers 67
6 Device 8, Function 0 Extended Configuration Test Overflow Registers 68
7 PCI to PCI Bridge Registers 79
8 LPC I/F Device 31, Function 0 Registers 80
9 CPU I/F Registers 81
10 TCO I/O Registers 82

11 GPIO Registers 83
12 USB I/O Registers 84
13 SMBUS Device 31, Function 3 Registers 85
14 SMB I/O Registers 86
15 USB 2.0 Registers 87
16 PCI Express Header Registers 89
17 SATA Registers 90
18 Concept Diagram of EDMA Data Path 111
19 Conceptual Diagram of Four Channel EDMA Engine 113
20 Chain Descriptor in Memory 115
21 Chaining Mechanism 116
22 Source and Destination in Increment Mode Transfer 122
23 Source in Decrement and Destination in Increment Mode Transfer (Byte Reversal) 123
24 Source in Increment and Destination in 1-Byte Granularity Constant Mode Transfer 124
25 Source in Increment and Destination in 2-Byte Granularity Constant Mode Transfer 125
26 Source in Increment and Destination in 4-Byte Granularity Constant Mode Transfer 126
27 Source in Decrement and Destination in 1-Byte Granularity Constant Mode Transfer 127
28 Source in Decrement and Destination in 2-Byte Granularity Constant Mode Transfer 128
29 Source in Decrement and Destination in 4-Byte Granularity Constant Mode Transfer 129
30 Source in Memory Initialization and Destination in Increment Mode Transfer 130
31 Source in Buffer Init and Destination in 1-Byte Granularity Constant Mode Transfer 131
32 Source in Buffer Init and Destination in 2-Byte Granularity Constant Mode Transfer 131
33 Source in Buffer Init and Destination in 4-Byte Granularity Constant Mode Transfer 132
34 Initiation Flow Chart 147
35 Completion Flow Chart 148
36 Intel® Pentium® M Processor Based Platform Power Delivery 151
37 Dual-Core Intel® Xeon® Processor LV Based Platform Power Delivery 152
38 IMCH's 1 x8 PCI Express* PEA Link Configuration Examples 161
39 IICH's 1 x4 PCI Express* Link Configuration Examples 161
40 Basic Memory Regions 177
41 DOS Legacy Region 178
42 Memory Region from 1 MByte through 4 GBytes 182
43 PAM Associated Attribute Bits 183
44 Bus 0 Device Map 203
45 NSI Type 0 Configuration Address Translation 205
46 NSI Type 1 Configuration Address Translation 205
47 Mechanism 1 Type 1 Configuration Address to PCI Address Mapping 206
48 IMCH Configuration Flow Chart 207
49 PCI Express* Configuration Transaction Header 210
50 Enhanced Configuration Memory Address Map 211
51 DDR2 Data Bursts 216
52 Global FERR/NERR Register Representation 217
53 FERR/NERR Service Routine 219
54 Dword Configuration Read Protocol 231
55 Dword Configuration Write Protocol 231
56 Dword Memory Read Protocol 232
57 Dword Memory Write Protocol 232
58 Dword Configuration Read Protocol 232
59 Dword Configuration Write Protocol 233
60 Dword Memory Read Protocol 233
61 WORD Configuration Write Protocol 234
62 PCI Express* Hot-Plug Overview 237
63 Hot-Plug Initialization Flow 238
64 Hot-Plug Insertion Flow Via Pushbutton Request 239
65 Hot-Plug Removal Flow Via Pushbutton 240

66 SMB Byte Write Transaction for Expander 241
67 SMB Byte Read Transaction for Expander 241
68 Platform Clocking 254
69 Power-On Reset Sequence 259
70 Package Specifications 263
71 Intel® 3100 Chipset Quadrant Layout (Top View) 264
72 Intel® 3100 Chipset Quadrant Layout (Bottom View) 265
73 Memory Channel Address Map 287
74 DDR2 CS Routing for Single- or Dual-rank DIMMS 288
75 DCAL Control and Status Register 487
76 Offset 104 - 107h: DCALADDR – DCAL Address Register 492
77 LPC Interface Diagram 596
78 LPC Bridge SERR# 600
79 IICH DMA Controller 601
80 DMA Request Assertion through LDRQ# 611
81 Coprocessor Error Timing Diagram 652
82 C0→C2→C0 Entry/Exit Timings 695
83 Example Queue Conditions 763
84 USB Data Encoding 766
85 USB Port Connections 876
86 SATA Power States 951
87 Hardware Flow for Port Enable/Device Present Bits 953
88 Port System Memory Structure 955
89 Port System Memory Structure 956
90 Power Management PME SMI/SCI/INTR Logic 1002
91 SERR# for Backbone Parity Errors 1003
92 SERR# for Root Port Parity Errors 1003
93 SERR# for Secondary Bus Errors 1004
94 SERR# for PCI Express Correctable Errors 1005
95 SERR# for PCI Express Uncorrectable (Non-Fatal) Errors 1005
96 SERR# for PCI Express Uncorrectable (Fatal) Errors 1005
97 Generation of SERR# to Platform 1006
98 Example UART Data Frame 1016
99 WDT Block Diagram 1032
100 Start Frame Timing with Source Sampled a Low Pulse on IRQ1 1042
101 Stop Frame Timing with Host Using Quiet Mode Sampling Period 1042

Tables

1 Glossary Table 39
2 Referenced Documents 47
3 Related Websites 47
4 Supported Microprocessors 50
5 DDR2-400 Memory Interface Capacities 51
6 Configuration Table Bit Types 63
7 IMCH Controller PCI Configuration Register Map (D0, F0) 69
8 Error Reporting PCI Configuration Register Map (D0, F1) 71
9 EDMA Configuration Register Map 72
10 PCI Express Port A Standard and Enhanced Configuration Register Map 73
11 Device 3, Function 0: PCI Express* Port A1 Standard and Enhanced Registers 75
12 Extended Configuration Register Map (D8,F0) 75
13 Memory Mapped I/O for DDR2 Register Summary 76
14 Memory Mapped I/O for EDMA Register Summary 76
15 Memory Mapped I/O Registers for NSI Register Summary 78
16 Bridging and Configuration Register (Memory Space) Summary Table 91

17 High Precision Event Timers Registers Summary 92
18 Patch Registers Summary Table 92
19 PCI to PCI Bridge Register Summary Table 93
20 Bridge Proprietary Configuration Registers Summary 95
21 LPC I/F – D31, F0 Configuration Registers Summary Table 95
22 DMA Registers Summary 96
23 8254 Timer Register Summary Table 97
24 APIC Indirect Registers (LPC I/F – D31, F0) Summary 97
25 APIC Direct Registers (LPC I/F – D31, F0) Summary 98
26 Processor I/F Registers Summary Table 98
27 Indexed Registers Summary 98
28 Summary Table for Power Management PCI Registers (PM — D31:F0) 98
29 TCO I/O Registers Summary Table 99
30 GPIO Register Summary Table 99
31 USB I/O Registers Summary 100
32 SMBUS Device 3, Function 3 Configuration Registers Summary 100
33 SMB I/O Registers Summary 100
34 SMB Slave Interface I/O Registers Summary 101
35 USB 2.0 Configuration Registers Summary Table 101
36 Host Controller Capability Registers Summary Table 103
37 Host Controller Operational Registers Summary Table 103
38 Debug Port Registers Summary Table 103
39 I/O Registers Summary Table 104
40 SATA PCI Header Registers Summary Table 104
41 SATA Register Summary: Additional SFF-8038i Configuration Registers 104
42 SATA Register Summary: PCI Power Management Capabilities Registers 105
43 SATA Register Summary: Message Signaled Interrupt Registers 105
44 SATA Additional Configuration Registers Summary 105
45 Generic Host Controller Register Summary Table 105
46 PCI Express Header Registers Summary Table 106
47 PCI Express Register Summary: Root Port Capability Structure 106
48 PCI Express Register Summary: Message Signaled Interrupt Capability 106
49 PCI Express Register Summary: PCI Bridge Vendor Capability 107
50 PCI Express Register Summary: PCI Power Management Capability 107
51 PCI Express Register Summary: Port Configuration Capability 107
52 PCI Express Register Summary: VC Configuration Capability 107
53 PCI Express Register Summary: Advanced Error Reporting Capability 107
54 PCI Express Register Summary: Root Complex Topology Capability Structure Registers ... 108
55 I/O Register Summary Table 108
56 Configuration Register Summary 108
57 Channel 0 Memory-mapped Register Set 136
58 Interrupt Summary 142
59 Terminology 149
60 Summary of Interfaces 149
61 Power Supply Requirements 150
62 Power Wells 150
63 Power and Ground Planes 153
64 CMOS and SCHMITT Signals Types 155
65 Processor Bus Interface 156
66 Processor Interface 159
67 DDR2 Memory Bus Interface 160
68 PCI Express* Interface 162
69 SMBus Interface 164
70 Clocks, Resets, and Miscellaneous Signals 164
71 LPC and FWH Interfaces 166

72 USB Interface 166
73 SATA Interface 167
74 UART Interface 167
75 Interrupt Interface 169
76 System Management and Power State Signals 169
77 Watchdog Timer and Real Time Clock Interfaces 170
78 PCI Interface 171
79 General Purpose I/O Interface 173
80 TAP and Debug Interface 174
81 Strapping Information 175
82 Regions of Memory Ranges 176
83 System Memory Space 177
84 IMCH VGA and MDA Memory Spaces 178
85 IMCH PAM Memory Address Ranges 180
86 PAM Associated Attribute Bits 184
87 ISA Hole Memory Space 184
88 TSEG SMM Memory Space 185
89 PCI Express* Enhanced Configuration Aperture 185
90 IOAPIC Memory Space 186
91 FSB Interrupt Memory Space 186
92 High SMM Memory Space 187
93 Device 2 Memory and Prefetchable Memory 188
94 Device 3 Memory and Prefetchable Memory 188
95 EDMA Accesses to Fixed Address Spaces 189
96 EDMA Accesses to Relocatable Address Spaces 189
97 Supported SMM Ranges 193
98 Fixed I/O Ranges Decoded by IICH 194
99 Variable I/O Decode Ranges 196
100 IICH Memory Decode Ranges (from CPU Perspective) 197
101 PCI Devices and Functions on Bus 0 201
102 CONFIG_ADDRESS - Configuration Address Register 209
103 CONFIG_DATA Configuration Data Register 209
104 Enhanced Configuration FSB Address Format 212
105 FSB Matrix Parity Scheme 213
106 Pseudocode for EDMA Errors 224
107 SMBus Register Summary 227
108 SMBUS Memory-Mapped Register Summary 227
109 ADDR3 Memory Assignments 227
110 Command (CMD) Register 228
111 Byte Count Register 229
112 Address Byte 3 Register 229
113 ADDR2 – Address Byte 2 Register 230
114 ADDR1 – Address Byte 1 Register 230
115 ADDR0 – Address Byte 0 Register 230
116 Offset 04-07: DATA - Data Register 230
117 Status Register 231
118 Slot Signal Legend 237
119 Single Byte Register Stack 241
120 External Single-Byte I/O Expander Event Matrix 242
121 Native x8 Personalities After BOOT 243
122 Relationship Between Link and Device PM States 246
123 Clocking Interfaces 252
124 CPU_SEL Divisor Selection 253
125 IICH and System Clock Domains 255
126 IMCH Reset Classes 256

127 Reset Sequences and Durations 259
128 Signal Listing (Sheet 1 of 7) 266
129 Signal Listing — Numerically by Ball Number (Sheet 1 of 7) 272
130 System Bus Package Trace Length (Sheet 1 of 5) 280
131 DDR2-400 Memory Interface Capacities 286
132 Supported DIMM Populations 287
133 Supported DDR2-400 DIMM Populations 288
134 DDR2 ODTC Settings for DDR2 289
135 DRA Mapping for DQS 290
136 DQS to DQ Mapping 290
137 Supported DDR2/FSB Clock Gearing Ratios 291
138 Supported DRAM Timings 291
139 IMCH Controller PCI Configuration Register Map (D0, F0) 293
140 Offset 00 - 01h: VID – Vendor Identification Register 295
141 Offset 02 - 03h: DID – Device Identification Register 295
142 Offset 04 - 05h: PCICMD – PCI Command Register 295
143 Offset 06 - 07h: PCISTS – PCI Status Register 297
144 Offset 08h: RID – Revision Identification Register 297
145 Offset 0Ah: SUBC – Sub-Class Code Register 298
146 Offset 0Bh: BCC – Base Class Code Register 298
147 Offset 0Dh: MLT – Master Latency Timer Register 299
148 Offset 0Eh: HDR – Header Type Register 299
149 Offset 14 - 17h: SMRBASE – System Memory RCOMP Base Address Register 299
150 Offset 2C - 2Dh: SVID – Subsystem Vendor Identification Register 300
151 Offset 2E - 2Fh: SID – Subsystem Identification Register 300
152 Offset 4C - 4Fh: NSIBAR – Root Complex Block Address Register 301
153 Offset 50h: IMCH CFG0 – IMCH Configuration 0 Register 301
154 Offset 51h: IMCH CFG1 – IMCH Configuration 1 Register 302
155 Offset 52h: IMCH CFGNS0 – IMCH Configuration 0 Register 302
156 Offset 53h: IMCH CFGNS1 – IMCH Configuration 1 Register 303
157 Offset 58h: FDHC – Fixed DRAM Hole Control Register 303
158 Offset 59h: PAM0 – Programmable Attribute Map 0 Register 304
159 Offset 5Ah: PAM1 – Programmable Attribute Map 1 Register 304
160 Offset 5Bh: PAM2 – Programmable Attribute Map 2 Register 305
161 Offset 5Ch: PAM3 – Programmable Attribute Map 3 Register 306
162 Offset 5Dh: PAM4 – Programmable Attribute Map 4 Register 306
163 Offset 5Eh: PAM5 – Programmable Attribute Map 5 Register 308
164 Offset 5Fh: PAM6 – Programmable Attribute Map 6 Register 309
165 Offset 60h: DRB0 – DRAM Row 0 Boundary Register 309
166 DRB to DIMM Designation 310
167 Example Configuration 310
168 Example Register Settings 310
169 Offset 61h: DRB1 – DRAM Row 1 Boundary Register 311
170 Offset 62h: DRB2 – DRAM Row 2 Boundary Register 311
171 Offset 63h: DRB3 – DRAM Row 3 Boundary Register 311
172 Offset 64h: DRB4 – DRAM Row 4 Boundary Register 312
173 Offset 65h: DRB5 – DRAM Row 5 Boundary Register 312
174 Offset 66h: DRB6 – DRAM Row 6 Boundary Register 312
175 Offset 67h: DRB7 – DRAM Row 7 Boundary Register 312
176 Offset 70h: DRA0 – DRAM Row 0 Attribute Register 313
177 DRA to DIMM Designation 313
178 Offset 71h: DRA1 – DRAM Row 1 Attribute Register 314
179 Offset 72h: DRA2 – DRAM Row 2 Attribute Register 314
180 Offset 73h: DRA3 – DRAM Row 3 Attribute Register 314
181 Offset 78 - 7Bh: DRT – DRAM Timing Register 315

182 Offset 7C - 7Fh: DRC – DRAM Controller Mode Register 320
183 Offset 80 - 81h: DRM – DRAM Mapping Register 323
184 Offset 82h: DRORC – Opportunistic Refresh Control Register 324
185 Offset 84 - 87h: ECCDIAG – ECC Detection/Correction Diagnostic Register 325
186 Offset 88 - 8Bh: SDRC – Secondary DRAM Controller Mode Register 326
187 Offset 8Ch: CKDIS – CK/CK# Clock Disable Register 328
188 Offset 8Dh: CKEDIS – CKE/CKE# Clock Disable Register 328
189 Offset 90 - 93h: SPARECTL – SPARE Control Register 329
190 Offset 94 - 97h: DRAMISCTL – DRAM Miscellaneous Control Register 330
191 Offset 9A - 9Bh: DDRCSR – DDR Channel Configuration Control/Status Register 331
192 Offset 9Ch: DEVPRES – Device Present Register 332
193 Offset 9Dh: EXSMRC – Extended System Management RAM Control Register 332
194 Offset 9Eh: SMRAM – System Management RAM Control Register 334
195 Offset 9Fh: EXSMRAMC – Expansion System Management RAM Control Register 336
196 Offset A0 - A3h: CLKGRFM0 – Clock Gearing Ratio FSB to Memory 0 Register 336
197 Offset A4 - A7h: CLKGRFM1 – Clock Gearing Ratio FSB to Memory 1 Register 337
198 Offset A8 - ABh: CLKGRMF0 – Clock Gearing Ratio Memory to FSB 0 Register 337
199 Offset AC - AFh: CLKGRMF1 – Clock Gearing Ratio Memory to FSB 1 Register 338
200 Offset B0 - B3h: DDR2ODTC – DDR2 ODT Control Register 338
201 Offset C4 - C5h: TOLM – Top of Low Memory Register 340
202 Offset C6 - C7h: REMAPBASE – Remap Base Address Register 340
203 Offset: C8 - C9h: REMAPLIMIT – Remap Limit Address Register 341
204 Offset CA - CBh: REMAPOFFSET – Remap Offset Register 341
205 Offset CC - CDh: TOM – Top Of Memory Register 342
206 Offset CE - CFh: HECBASE – PCI Express Port A (PEA)
Enhanced Configuration Base Address Register 342
207 Offset D8h: CACHECTL0 – Write Cache Control 0 Register 343
208 Offset DE - DFh: SKPD – Scratchpad Data Register 343
209 Offset F5h: IMCH TST1 – IMCH Test Byte 1 Register 343
210 Error Reporting PCI Configuration Register Map (D0, F1) 344
211 Offset 00 - 01h: VID – Vendor Identification Register 346
212 Offset 02 - 03h: DID – Device Identification Register 346
213 Offset 04 - 05h: PCICMD – PCI Command Register 347
214 Offset 06 - 07h: PCISTS – PCI Status Register 347
215 Offset 08h: RID – Revision Identification Register 348
216 Offset 0Ah: SUBC – Sub-Class Code Register 348
217 Offset 0Bh: BCC – Base Class Code Register 348
218 Offset 0Eh: HDR – Header Type Register 349
219 Offset 2C - 2Dh: SVID – Subsystem Vendor Identification Register 349
220 Offset 2E - 2Fh: SID – Subsystem Identification Register 349
221 Offset 40 - 43h: GLOBAL_FERR – Global First Error Register 350
222 Offset 44 - 47h: GLOBAL_NERR – Global Next Error Register 351
223 Offset 48 - 4Bh: NSI_FERR – NSI First Error Register 352
224 Offset 4C - 4Fh: NSI_NERR – NSI Next Error Register 354
225 Offset 50 - 53h: NSI_SCICMD – NSI SCI Command Register 355
226 Offset 54 - 57h: NSI_SMICMD – NSI SMI Command Register 357
227 Offset 58 - 5Bh: NSI_SERRCMD – NSI SERR Command Register 359
228 Offset 5C - 5Fh: NSI_MCERRCMD – NSI MCERR Command Register 361
229 Offset 60 - 61h: FSB_FERR – FSB First Error Register 363
230 Offset 62 - 63h: FSB_NERR – FSB Next Error Register 364
231 Offset 64 - 65h: FSB_EMASK – FSB Error Mask Register 364
232 Offset 68 - 69h: FSB_SCICMD – FSB SCI Command Register 365
233 Offset 6A - 6Bh: FSB_SMICMD – FSB SMI Command Register 367
234 Offset 6C - 6Dh: FSB_SERRCMD – FSB SERR Command Register 368
235 Offset 6E - 6Fh: FSB_MCERRCMD – FSB MCERR Command Register 370

236 Offset 72h: BUF_NERR – Memory Buffer Next Error Register 371
237 Offset 74h: BUF_EMASK – Memory Buffer Error Mask Register 372
238 Offset 78h: BUF_SCICMD – Memory Buffer SCI Command Register 372
239 Offset 7Ah: BUF_SMICMD – Memory Buffer SMI Command Register 373
240 Offset 7Ch: BUF_SERRCMD – Memory Buffer SERR Command Register 374
241 Offset 7Eh: BUF_MCERRCMD – Memory Buffer MCERR Command Register 375
242 Offset 80 - 81h: DRAM_FERR – DRAM First Error Register 376
243 Offset 82 - 83h: DRAM_NERR – DRAM Next Error Register 377
244 Offset 84h: DRAM_EMASK – DRAM Error Mask Register 377
245 Offset 88h: DRAM_SCICMD – DRAM SCI Command Register 378
246 Offset 8AH: DRAM_SMICMD – DRAM SMI Command Register 379
247 Offset 8Ch: DRAM_SERRCMD – DRAM SERR Command Register 380
248 Offset 8Eh: DRAM_MCERRCMD – DRAM MCERR Command Register 381
249 Offset 98 - 99h: THRESH_SEC0 – DIMM0 SEC Threshold Register 382
250 Offset 9A - 9Bh: THRESH_SEC1 – DIMM1 SEC Threshold Register 382
251 Offset 9C - 9Dh: THRESH_SEC2 – DIMM2 SEC Threshold Register 383
252 Offset 9E - 9Fh: THRESH_SEC3 – DIMM3 SEC Threshold Register 383
253 Offset A0 - A3h: DRAM_SECF_ADD – DRAM First Single Bit Error Correct Address Register 383
254 Offset A4 - A7h: DRAM_DED_ADD – DRAM Double Bit Error Address Register 384
255 Offset A8 - ABh: DRAM_SCRB_ADD – DRAM Scrub Error Address Register 384
256 Offset AC - AFh: DRAM_RETR_ADD – DRAM DED Retry Address Register 385
257 Offset B0 - B1h: DRAM_SEC_D0A – DRAM SEC Logical DIMM 0 Counter Register 385
258 Offset B2 - B3h: DRAM_DED_D0A – DRAM DED Logical DIMM 0 Counter Register 386
259 Offset B4 - B5h: DRAM_SEC_D1A – DRAM SEC Logical DIMM 1 Counter Register 386
260 Offset B6 - B7h: DRAM_DED_D1A – DRAM DED Logical DIMM 1 Counter Register 386
261 Offset B8 - B9h: DRAM_SEC_D2A – DRAM SEC Logical DIMM 2 Counter Register 387
262 Offset BA - BBh: DRAM_DED_D2A – DRAM DED Logical DIMM 2 Counter Register 387
263 Offset BC - BDh: DRAM_SEC_D3A – DRAM SEC Logical DIMM 3 Counter Register 387
264 Offset BE - BFh: DRAM_DED_D3A – DRAM DED Logical DIMM 3 Counter Register 388
265 Offset C2 - C3h: THRESH_DED – Threshold for DEDs Register 388
266 Offset C4 - C5h: DRAM_SECF_SYNDROME – DRAM First Single Error Correct Syndrome Register 389
267 Offset C6 - C7h: DRAM_SECN_SYNDROME – DRAM Next Single Error Correct Syndrome Register 389
268 Offset C8 - CBh: DRAM_SECN_ADD – DRAM Next Single Bit Error Correct Address Register 390
269 Offset DC - DDh: DIMMTHREX – DIMM Threshold Exceeded Register 390
270 Offset E0h - E3h: HERRCTL – Host Error Control Register 391
271 Offset E8h - EBh: BERRCTL – Buffer Error Control Register 392
272 Offset ECh - EFh: DERRCTL – DRAM Error Control Register 393
273 EDMA Configuration Register Map 394
274 Offset 00h - 01h: VID – Vendor Identification Register 395
275 Offset 02h - 03h: DID – Device Identification Register 395
276 Offset 04h - 05h: PCICMD – PCI Command Register 395
277 Offset 06h - 07h: PCISTS – PCI Status Register 396
278 Offset 08h: RID – Revision Identification Register 397
279 Offset 0Ah: SUBC – Sub-Class Code Register 397
280 Offset 0Bh: BCC – Base Class Code Register 398
281 Offset 0Eh: HDR – Header Type Register 398
282 Offset 10h - 13h: EDMALBAR – EDMA Low Base Address Register 398
283 Offset 2Ch - 2Dh: SVID – Subsystem Vendor Identification Register 399
284 Offset 2Eh - 2Fh: SID – Subsystem Identification Register 399
285 Offset 34h: CAPPTR – Capabilities Pointer Register 399
286 Offset 3Ch: INTRLINE – Interrupt Line Register 400

287 Offset 3Dh: INTRPIN – Interrupt Pin Register 400
288 Offset 40h: EDMACTL – EDMA Control Register 400
289 Offset 80h - 83h: EDMA_FERR – EDMA First Error Register 401
290 Offset 84h - 87h: EDMA_NERR – EDMA Next Error Register 402
291 Offset 88h: EDMA_EMASK – EDMA Error Mask Register 404
292 Offset A0h: EDMA_SCICMD – EDMA SCI Command Register 405
293 Offset A4h: EDMA_SMICMD – EDMA SMI Command Register 406
294 Offset A8h: EDMA_SERRCMD – EDMA SERR Command Register 407
295 Offset ACh: EDMA_MCERRCMD – EDMA MCERR Command Register 408
296 Offset B0h - B3h: MSICR – MSI Control Register 409
297 Offset B4h - B7h: MSIAR – MSI Address Register 410
298 Offset B8h - B9h: MSIDR – MSI Data Register 411
299 PCI Express Port A Standard and Enhanced Configuration Register Map 413
300 Offset 00 - 01h: VID – Vendor Identification Register 415
301 Offset 02 - 03h: DID – Device Identification Register 415
302 Offset 04 - 05h: PCICMD – PCI Command Register 415
303 Offset 06 - 07h: PCISTS – PCI Status Register 417
304 Offset 08h: RID – Revision Identification Register 418
305 Offset 0Ah: SUBC – Sub-Class Code Register 418
306 Offset 0Bh: BCC – Base Class Code Register 419
307 Offset 0Ch: CLS – Cache Line Size Register 419
308 Offset 0Eh: HDR – Header Type Register 419
309 Offset 18h: PBUSN – Primary Bus Number Register 420
310 Offset 19h: SBUSN – Secondary Bus Number Register 420
311 Offset 1Ah: SUBUSN – Subordinate Bus Number Register 420
312 Offset 1Ch: IOBASE – I/O Base Address Register 421
313 Offset 1Dh: IOLIMIT – I/O Limit Address Register 421
314 Offset 1E - 1Fh: SECSTS – Secondary Status Register 422
315 Offset 20 - 21h: MBASE – Memory Base Address Register 423
316 Offset 22 - 23h: MLIMIT – Memory Limit Address Register 424
317 Offset 24 - 25h: PMBASE – Prefetchable Memory Base Address Register 425
318 Offset 26 - 27h: PMLIMIT – Prefetchable Memory Limit Address Register 425
319 Offset 28h: PMBASU – Prefetchable Memory Base Upper Address Register 426
320 Offset 2Ch: PMLMTU – Prefetchable Memory Limit Upper Address Register 426
321 Offset 34h: CAPPTR – Capabilities Pointer Register 426
322 Offset 3Ch: INTRLINE – Interrupt Line Register 427
323 Offset 3Dh: INTRPIN – Interrupt Pin Register 427
324 Offset 3Eh: BCTRL – Bridge Control Register 427
325 Offset 44h: VSCMD0 – Vendor Specific Command Byte 0 Register 429
326 Offset 45h: VSCMD1 – Vendor Specific Command Byte 1 Register 430
327 Offset 46h: VSSTS0 – Vendor Specific Status Byte 0 Register 430
328 Offset 47h: VSSTS1 – Vendor Specific Status Byte 1 Register 431
329 Offset 50h: PMCAPID – Power Management Capabilities Structure Register 432
330 Offset 51h: PMNPTR – Power Management Next Capabilities Pointer Register 432
331 Offset 52 - 53h: PMCAPA – Power Management Capabilities Register 433
332 Offset 54 - 55h: PMCSR – Power Management Status and Control Register 433
333 Offset 56h: PMCSRBSE – Power Management Status and Control Bridge Extensions Register 434
334 Offset 58h: MSICAPID – MSI Capabilities Structure Register 434
335 Offset 59h: MSINPTR – MSI Next Capabilities Pointer Register 435
336 Offset 5A - 5Bh: MSICAPA – MSI Capabilities Register 435
337 Offset 5C - 5Fh: MSIAR – MSI Address for PCI Express Register 436
338 Offset 60 - 61h: MSIDR – MSI Data Register 436
339 Offset 64h: PEACAPID – PCI Express Features Capabilities ID Register 437
340 Offset 65h: PEANPTR – PCI Express Next Capabilities Pointer Register 437

341 Offset 66 - 67h: PEACAPA – PCI Express Features Capabilities Register 438
342 Offset 68 - 6Bh: PEADEVCAP – PCI Express Device Capabilities Register 438
343 Offset 6C - 6Dh: PEADEVCTL – PCI Express Device Control Register 439
344 Offset 6E - 6Fh: PEADEVSTS – PCI Express Device Status Register 441
345 Offset 70 - 73h: PEALNKCAP – PCI Express Link Capabilities Register 442
346 Offset 74 - 75h: PEALNKCTL – PCI Express Link Control Register 443
347 Offset 76 - 77h: PEALNKSTS – PCI Express Link Status Register 444
348 Offset 78 - 7Bh: PEASLTCAP – PCI Express Slot Capabilities Register 445
349 Offset 7C - 7Dh: PEASLTCTL – PCI Express Slot Control Register 446
350 Offset 7E - 7Fh: PEASLTSTS – PCI Express Slot Status Register 448
351 Offset 80 - 83h: PEARPCTL – PCI Express Root Port Control Register 449
352 Offset 84 - 87h: PEARPSTS – PCI Express Root Port Status Register 450
353 Offset 100 - 103h: ENHCAPST – Enhanced Capability Structure Register 450
354 Offset 104 - 107h: UNCERRSTS – Uncorrectable Error Status Register 451
355 Offset 108 - 10Bh: UNCERRMSK – Uncorrectable Error Mask Register 453
356 Offset 10C - 10Fh: UNCERRSEV – Uncorrectable Error Severity Register 454
357 Offset 110 - 113h: CORERRSTS – Correctable Error Status Register 455
358 Offset 114 - 117h: CORERRMSK – Correctable Error Mask Register 456
359 Offset 118 - 11Bh: AERCACR – Advanced Error Capabilities and Control Register 457
360 Offset 11C - 11Fh: HDRLOG0 – Header Log DW 0 (1st 32 bits) Register 458
361 Offset 120 - 123h: HDRLOG1 – Header Log DW 1 (2nd 32 bits) Register 458
362 Offset 124 - 127h: HDRLOG2 – Header Log DW 2 (3rd 32 bits) Register 458
363 Offset 128 - 12Bh: HDRLOG3 – Header Log DW 3 (4th 32 bits) Register 459
364 Offset 12C - 12Fh: RPERRCMD – Root (Port) Error Command Register 459
365 Offset 130 - 133h: RPERRMSTS – Root (Port) Error Message Status Register 460
366 Offset 134 - 137h: ERRSID – Error Source ID Register 461
367 Offset 140 - 143h: PEAUNITERR – PCI Express Unit Error Register 462
368 Offset 144 - 147h: PEAMASKERR – PCI Express Unit Mask Error Register 464
369 Offset 148 - 14Bh: PEAERRDOCMD – PCI Express Error Do Command Register 466
370 Offset 14C - 14Fh: UNCEDMASK – Uncorrectable Error Detect Mask Register 468
371 Offset 150 - 153h: COREDMASK – Correctable Error Detect Mask Register 469
372 Offset 158 - 15Bh: PEAUNITEDMASK – PCI Express Unit Error Detect Mask Register 469
373 Offset 160 - 163h: PEAFERR – PCI Express First Error Register 471
374 Offset 164 - 167h: PEANERR – PCI Express Next Error Register 472
375 Offset 168 - 16Bh: PEAERRCTL – PCI Express Port A Error Control Register 473
376 Device 3, Function 0: PCI Express* Port A1 Standard and Enhanced Registers 474
377 Offset 02 - 03h: DID – Device Identification Register 474
378 Offset 70 - 73h: PEA1LNKCAP – PCI Express Link Capabilities Register 475
379 Extended Configuration Register Map (D8,F0) 476
380 Offset 00 - 01h: VID – Vendor Identification Register 476
381 Offset 02 - 03h: DID – Device Identification Register 477
382 Offset 04 - 05h: PCICMD – PCI Command Register 477
383 Offset 06 - 07h: PCISTS – PCI Status Register 478
384 Offset 08h: RID – Revision Identification Register 478
385 Offset 0Ah: SUBC – Sub-Class Code Register 479
386 Offset 0Bh: BCC – Base Class Code Register 479
387 Offset 0Eh: HDR – Header Type Register 479
388 Offset 2C - 2Dh: SVID – Subsystem Vendor Identification Register 480
389 Offset 2E - 2Fh: SID – Subsystem Identification Register 480
390 Offset B6 - B7h: HPCCTL – Hot Plug Controller Control Register 481
391 Offset C8 - CBh: SCRUBLIM – Scrub Limit and Other Control Information Register 481
392 Offset CC - CFh: SCRBADD – Scrub Address Register 482
393 Offset D0 - D3h: DTCL – DRAM Throttling Control Lower Register 483
394 Offset D4 - D7h: DTCU – DRAM Throttling Control Upper Register 484
395 Memory Mapped I/O for DDR2 Register Summary 485

396 Offset 00 - 01h: NOTESPAD – Note (Sticky) Pad for BIOS Support Register 486
397 Offset 02 - 03h: NOTEPAD – Note Pad for BIOS Support Register 486
398 Offset 100 - 103h: DCALCSR – DCAL Control and Status Register 487
399 Offset 104 - 107h: DCALADDR – DCAL Address Register 493
400 Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0 494
401 Offset 10C - 10Fh: DCALDATA1 – DCAL Data Register DW1 501
402 Offset 110 - 113h: DCALDATA2 – DCAL Data Register DW2 501
403 Offset 114 - 117h: DCALDATA3 – DCAL Data Register DW3 501
404 Offset 118 - 11Bh: DCALDATA4 – DCAL Data Register DW4 502
405 Offset 11C - 11Fh: DCALDATA5 – DCAL Data Register DW5 502
406 Offset 120 - 123h: DCALDATA6 – DCAL Data Register DW6 502
407 Offset 124 - 127h: DCALDATA7 – DCAL Data Register DW7 503
408 Offset 128 - 12Bh: DCALDATA8 – DCAL Data Register DW8 503
409 Offset 12C - 12Fh: DCALDATA9 – DCAL Data Register DW9 503
410 Offset 130 - 133h: DCALDATA10 – DCAL Data Register DW10 504
411 Offset 134 - 137h: DCALDATA11 – DCAL Data Register DW11 504
412 Offset 138 - 13Bh: DCALDATA12 – DCAL Data Register DW12 504
413 Offset 13C - 13Fh: DCALDATA13 – DCAL Data Register DW13 505
414 Offset 140 - 143h: DCALDATA14 – DCAL Data Register DW14 505
415 Offset 144 - 147h: DCALDATA15 – DCAL Data Register DW15 505
416 Offset 148 - 14Bh: DCALDATA16 – DCAL Data Register DW16 506
417 Offset 14C - 14Fh: DCALDATA17 – DCAL Data Register DW17 506
418 Offset 150 - 153h: RCVENDLYA – Receive Enable Delay Register 506
419 Offset 200 - 203h: DQSOFCSA01L – DQS Offset to CS0&1 Lower Register 507
420 Offset 204 - 207h: DQSOFCSA01M – DQS Offset to CS0&1 Middle Register 507
421 Offset 208h: DQSOFCSA01U – DQS Offset to CS0&1 Upper Register 508
422 Offset 20C - 20Fh: DQSOFCSA23L – DQS Offset to CS2&3 Lower Register 508
423 Offset 210 - 213h: DQSOFCSA23M – DQS Offset to Channel A CS2&3 Middle Register 508
424 Offset 214h: DQSOFCSA23U – DQS Offset to CS2&3 Upper Register 509
425 Offset 218 - 21Bh: DQSOFCSA45L – DQS Offset to CS4&5 Lower Register 509
426 Offset 21C - 21Fh: DQSOFCSA45M – DQS Offset to CS4&5 Middle Register 509
427 Offset 220h: DQSOFCSA45U – DQS Offset to CS4&5 Upper Register 509
428 Offset 224 - 227h: DQSOFCSA67L – DQS Offset to A CS6&7 Lower Register 510
429 Offset 228 - 22Bh: DQSOFCSA67M – DQS Offset to CS6&7 Middle Register 510
430 Offset 22Ch: DQSOFCSA67U – DQS Offset to CS6&7 Upper Register 510
431 Memory Mapped I/O for EDMA Register Summary 511
432 Offset 00 - 03h: CCR0 – Channel 0 Channel Control Register 513
433 Offset 04 - 07h: CSR0 – Channel 0 Channel Status Register 514
434 Offset 08 - 0Bh: CDAR0 – Channel 0 Current Descriptor Address Register 515
435 Offset 0C - 0Fh: CDUAR0–Channel 0 Current Descriptor Upper Address Register 515
436 Offset 10 - 13h: SAR0 – Channel 0 Source Address Register 515
437 Offset 14 - 17h: SUAR0 – Channel 0 Source Upper Address Register 516
438 Offset 18 - 1Bh: DAR0 – Channel 0 Destination Address Register 516
439 Offset 1C - 1Fh: DUAR0 – Channel 0 Destination Upper Address Register 516
440 Offset 20 - 23h: NDAR0 – Channel 0 Next Descriptor Address Register 517
441 Offset 24 - 27h: NDUAR0 – Channel 0 Next Descriptor Upper Address Register 517
442 Offset 28 - 2Bh: TCR0 – Channel 0 Transfer Count Register 518
443 Offset 2C - 2Fh: DCR0 – Channel 0 Descriptor Control Register 518
444 Offset 40 - 43h: CCR1 – Channel 1 Channel Control Register 520
445 Offset 44 - 47h: CSR1 – Channel 1 Channel Status Register 520
446 Offset 48 - 4Bh: CDAR1 – Channel 1 Current Descriptor Address Register 521
447 Offset 4C - 4Fh: CDUAR1 – Channel 1 Current Descriptor Upper Address Register 521
448 Offset 50 - 53h: SAR1 – Channel 1 Source Address Register 521
449 Offset 54 - 57h: SUAR1 – Channel 1 Source Upper Address Register 521
450 Offset 58 - 5Bh: DAR1 – Channel 1 Destination Address Register 522

451 Offset 5C - 5Fh: DUAR1 – Channel 1 Destination Upper Address Register 522
452 Offset 60 - 63h: NDAR1 – Channel 1 Next Descriptor Address Register..... 522
453 Offset 64 - 67h: NDUAR1 – Channel 1 Next Descriptor Upper Address Register 522
454 Offset 68 - 6Bh: TCR1 – Channel 1 Transfer Count Register..... 523
455 Offset 6C - 6Fh: DCR1 – Channel 1 Descriptor Control Register 523
456 Offset 80 - 83h: CCR2 – Channel 2 Channel Control Register 523
457 Offset 84 - 87h: CSR2 – Channel 2 Channel Status Register 523
458 Offset 88 - 8Bh: CDAR2 – Channel 2 Current Descriptor Address Register..... 524
459 Offset 8C - 8Fh: CDUAR2 – Channel 2 Current Descriptor Upper Address Register..... 524
460 Offset 90 - 93h: SAR2 – Channel 2 Source Address Register 524
461 Offset 94 - 97h: SUAR2 – Channel 2 Source Upper Address Register..... 524
462 Offset 98 - 9Bh: DAR2 – Channel 2 Destination Address Register 525
463 Offset 9C - 9Fh: DUAR2 – Channel 2 Destination Upper Address Register 525
464 Offset A0 - A3h: NDAR2 – Channel 2 Next Descriptor Address Register 525
465 Offset A4 - A7h: NDUAR2 – Channel 2 Next Descriptor Upper Address Register 525
466 Offset A8 - ABh: TCR2 – Channel 2 Transfer Count Register..... 526
467 Offset AC - AFh: DCR2 – Channel 2 Descriptor Control Register..... 526
468 Offset C0 - C3h: CCR3 – Channel 3 Channel Control Register 526
469 Offset C4 - C7h: CSR3 – Channel 3 Channel Status Register 526
470 Offset C8 - CBh: CDAR3 – Channel 3 Current Descriptor Address Register 527
471 Offset CC - CFh: CDUAR3 – Channel 3 Current Descriptor Upper Address Register 527
472 Offset D0 - D3h: SAR3 – Channel 3 Source Address Register 527
473 Offset D4 - D7h: SUAR3 – Channel 3 Source Upper Address Register 527
474 Offset D8 - DBh: DAR3 – Channel 3 Destination Address Register..... 528
475 Offset DC - DFh: DUAR3 – Channel 3 Destination Upper Address Register..... 528
476 Offset E0 - E3h: NDAR3 – Channel 3 Next Descriptor Address Register..... 528
477 Offset E4 - E7h: NDUAR3 – Channel 3 Next Descriptor Upper Address Register 528
478 Offset E8 - EBh: TCR3 – Channel 3 Transfer Count Register..... 529
479 Offset EC - EFh: DCR3 – Channel 3 Descriptor Control Register 529
480 Offset 100 - 103h: DCGC – EDMA Controller Global Command 529
481 Offset 104 - 107h: DCGS – EDMA Controller Global Status 530
482 Bridging and Configuration Register (Memory Space) Summary Table 531
483 Offset 0000 - 0003h: VCH – Virtual Channel Capability Header Register 533
484 Offset 0004 - 0007h: VCAP1 – Virtual Channel Capability 1 Register 533
485 Offset 0008 - 000Bh: VCAP2 – Virtual Channel Capability 2 Register 533
486 Offset 000C - 000Dh: PVC – Port Virtual Channel Control Register 534
487 Offset 000E - 000Fh: PVS – Port Virtual Channel Status Register 534
488 Offset 0010 - 00013h: V0CAP – Virtual Channel 0 Resource Capability Register..... 534
489 Offset 0014 - 00017h: V0CTL – Virtual Channel 0 Resource Control Register..... 535
490 Offset 001A - 001Bh: V0STS – Virtual Channel 0 Resource Status Register 535
491 Offset 0100 - 0103h: RCTCL – Root Complex Topology Capabilities List Register 536
492 Offset 0104 - 0107h: ESD – Element Self Description Register 536
493 Offset 0110 - 0113h: ULD – Upstream Link Description Register..... 537
494 Offset 0118 - 011Fh: ULBA - Upstream Link Base Address Register..... 537
495 Offset 0120 - 0123h: RPB0D – Root Port B0 (PEB0) Description Register 537
496 Offset 0128 - 012Fh: RPB0BA – Root Port B0 (PEB0) Base Address Register 538
497 Offset 0130 - 0133h: RPB1D – Root Port B1(PEB1) Description Register 538
498 Offset 0138 - 013Fh: RPB1BA – Root Port B1 (PEB1) Base Address Register 539
499 Offset 0140 - 0143h: RPB2D – Root Port B2 (PEB2) Description Register 539
500 Offset 0148 - 014Fh: RPB2BA – Root Port B2 (PEB2) Base Address Register1..... 540
501 Offset 0150 - 0153h: RPB3D – Root Port B3 (PEB3) Description Register 540
502 Offset 0158 - 015Fh: RPB3BA – Root Port B3 (PEB3) Base Address Register 541
503 Offset 01A0 - 01A3h: ILCL – Internal Link Capabilities List Register 541
504 Offset 01A4 - 01A7h: LCAP – Link Capabilities Register 541
505 Offset 01A8 - 01A9h: LCTL – Link Control Register 542

506 Offset 01AA - 01ABh: LSTS – Link Status Register 542
507 Offset 0224 - 0227h: RPC – Root Port Configuration Register 543
508 Offset 3000h: TCTL – TCO Control Register 544
509 Offset 3100 - 3103h: D31IP – Device 31 Interrupt Pin Register 544
510 Offset 3104 - 3107h: D30IP – Device 30 Interrupt Pin Register 545
511 Offset 3108 - 310Bh: D29IP – Device 29 Interrupt Pin Register 545
512 Offset 310C - 310Fh: D28IP – Device 28 Interrupt Pin Register 546
513 Offset 3140 - 3141h: D31IR – Device 31 Interrupt Route Register 546
514 Offset 3142 - 3143h: D30IR – Device 30 Interrupt Route Register 547
515 Offset 3144 - 3145h: D29IR – Device 29 Interrupt Route Register 547
516 Offset 3146 - 3147h: D28IR – Device 28 Interrupt Route Register 548
517 Offset 31FFh: OIC – Other Interrupt Control Register 548
518 Offset 3400 - 3403h: RC – RTC Configuration Register 549
519 Offset 3404 - 3407h: HPTC – High Performance Precision Timer Configuration Register 549
520 Offset 3410 - 3413h: GCS – General Control and Status Register 550
521 Offset 3414h: BUC – Backed Up Control Register 552
522 Offset 3418 - 341Bh: FD – Function Disable Register 552
523 Offset 341C - 341Fh: PRC – Power Reduction Control Register Clock Gating 554
524 PCI to PCI Bridge Register Summary Table 555
525 Offset 00 - 03h: ID – Identifiers Register 556
526 Offset 04 - 05h: CMD – Command Register 556
527 Offset 06 - 07h: PSTS – Primary Status Register 557
528 Offset 08h: RID – Revision Identification Register 559
529 Offset 09 - 0Bh: CC – Class Code Register 560
530 Offset 0Dh: PMLT – Primary Latency Timer Register 560
531 Offset 0Eh: HEADTYP – Header Type Register 560
532 Offset 18 - 1Ah: BNUM – Bus Number Register 561
533 Offset 1Bh: SMLT – Secondary Master Latency Timer Register 561
534 Offset 1C - 1Dh: IOBASE_LIMIT – I/O Base and Limit Register 561
535 Offset 1E - 1Fh: SSTS – Secondary Status Register 562
536 Offset 20 - 23h: MEMBASE_LIMIT – Memory Base and Limit Register 563
537 Offset 24 - 27h: PREF_MEM_BASE_LIMIT – Prefetchable Memory Base and Limit Register 564
538 Offset 28 - 2Bh: PMBU32 – Prefetchable Memory Base Upper 32 Bit Register 564
539 Offset 2C - 2Fh: PMLU32 – Prefetchable Memory Limit Upper 32-Bit Register 564
540 Offset 34h: CAPP – Capabilities List Pointer Register 565
541 Offset 3C - 3Dh: INTR – Interrupt Information Register 565
542 Offset 3E - 3Fh: BCTRL – Bridge Control Register 565
543 Bridge Proprietary Configuration Registers Summary 567
544 Offset 40 - 41h: SPDH – Secondary PCI Device Hiding Register 568
545 Offset 44 - 47h: DTC – Delayed Transaction Control Register 569
546 Offset 48 - 4Bh: BPS – Bridge Proprietary Status Register 570
547 Offset 4C - 4Fh: BPC – Bridge Policy Configuration Register 571
548 Offset 50h: SVCAP – Subsystem Vendor Capability Register 572
549 Offset 54h: SVID – Subsystem Vendor IDs Register 573
550 Offset F8h: MANID – Manufacturer's ID Register 573
551 PCI Bridge Initiator Cycle Types 573
552 LPC I/F – D31, F0 Configuration Registers Summary Table 579
553 Offset 00 - 03h: ID – Vendor Identification Register 580
554 Offset 04 - 05h: CMD – Device Command Register 581
555 Offset 06 - 07h: STS – Status Register 581
556 Offset 08h: RID – Revision ID Register 582
557 Offset 09 - 0Bh: CC – Class Code Register 583
558 Offset 0Dh: MLT – Master Latency Timer Register 583
559 Offset 0Eh: HTYPE – Header Type Register 583

560 Offset 2C - 2Fh: SID – Subsystem Identifiers Register 584
561 Offset 40 - 43h: ABASE – ACPI Base Address Register 584
562 Offset 44 - 47h: ACTL – ACPI Control Register 585
563 Offset 48h: GBA – GPIO Base Address Register 585
564 Offset 4Ch: GC – GPIO Control Register 586
565 Offset 60h: PARC – PIRQ[A,B,C,D,E,F,G,H] Routing Control Register 586
566 Offset 64h: SCNT – Serial IRQ Control Register 587
567 Offset 80 - 81h: IOD – I/O Decode Ranges Register 587
568 Offset 82 - 83h: IOE – I/O Enables Register 589
569 Offset 84 - 85h: LG1 – LPC Generic Decode Range 1 Register 590
570 Offset 88h: LG2 – LPC Generic Decode Range 2 Register 590
571 Offset D0 - D3h: FS1 – FWH ID Select 1 Register 591
572 Offset D4 - D5h: FS2 – FWH ID Select 2 Register 592
573 Offset D8 - DBh: FDE – FWH Decode Enable Register 592
574 Offset DCh: BC – BIOS Control Register 594
575 Offset F0h: RCBA – Root Complex Base Address Register 595
576 Offset F8h: MANID – Manufacturer's ID 595
577 LPC Cycle Types Supported 597
578 DMA Registers Summary 602
579 DMA Base and Current Address Registers 603
580 DMA Base and Current Count Registers 604
581 DMA Command Register 604
582 DMA Memory Low Page Registers 605
583 DMA Status Register 605
584 DMA Write Single Mask Register 606
585 DMA Channel Mode Register 606
586 DMA Clear Byte Pointer Register 607
587 DMA Master Clear Register 608
588 DMA Clear Mask Register 608
589 DMA Write All Mask Register 608
590 DMA Channel Priority 609
591 Address Shifting in 16-bit DMA Transfers 610
592 SPKR Signal 615
593 8254 Timer Register Summary Table 615
594 Counter Operating Modes 616
595 Offset 43h: TCW – Timer Control Word Register 617
596 Counter Latch Command 619
597 Read Back Command 620
598 Interval Timer Status Byte Format Register 621
599 Counter Access Ports Register 621
600 Interrupt Options - 8259 Mode 622
602 Signals Associated with Interrupt Logic 623
601 Interrupt Options - APIC Mode 623
603 8259 Core Connection 624
604 8259 Interrupt Controller (PIC) Registers (LPC I/F – D31, F0) 625
605 ICW1 – Initialization Command Word 1 626
606 Offset 21h/A1h: ICW2 – Initialization Command Word 2 626
607 Offset 21h: MICW3 – Master Initialization Command Word 3 627
608 Offset A1h: SICW3 – Slave Initialization Command Word 3 627
609 Offset 21h/A1h: ICW4 – Initialization Command Word 4 Register 628
610 OCW1 – Operational Control Word 1 (Interrupt Mask) 628
611 OCW2 – Operational Control Word 2 629
612 OCW3 – Operational Control Word 3 629
613 Offset 4D0h: ELCR1 – Master Edge/Level Control 630
614 Offset 4D1h: ELCR2 – Slave Edge/Level Control 631

615 Interrupt Handling 632
616 Content of Interrupt Vector Byte 632
617 Interrupt Delivery Address Format 638
618 Interrupt Delivery Data Format 639
619 APIC Direct Registers (LPC I/F – D31, F0) Summary 639
620 Address FEC00000h: IDX – Index Register 639
621 Address FEC00010h: DAT – Data Register 640
622 Address FEC00040h: EOI – EOI Register 640
623 APIC Indirect Registers (LPC I/F – D31, F0) Summary 640
624 Offset 00h: ID – Identification Register 641
625 Offset 01h: VS – Version Register 641
626 Offset 10-11h – 3E-3Fh: RTE[0-23] – Redirection Table Entry 642
627 Stop Frame Definition 645
628 Data Frame Format 645
629 Processor I/F Signal State 648
630 Processor I/F Registers Summary Table 648
631 Offset 61h: NMI_STS_CNT – NMI Status and Control Register 649
632 Offset 70h: NMI_EN – NMI Enable (and Real Time Clock Index) Register 650
633 Offset 92h: PORT92 – Fast A20 and Init Register 650
634 Offset F0h: COPROC_ERR – Coprocessor Error Register 650
635 Offset CF9h: RST_CNT – Reset Control Register 651
636 INIT# Going Active 652
637 NMI Sources 653
638 I/O Registers 655
639 RTC (Standard) RAM Bank 655
640 Indexed Registers Summary 656
641 RTC_REGA – Register A (General Configuration) 656
642 RTC_REGB – Register B (General Configuration) 658
643 RTC_REGC – Register C (Flag Register) 659
644 RTC_REGD – Register D (Flag Register) 659
645 General Power States and Consumption for Systems 662
646 State Transition Rules 662
647 System Power Planes 663
648 IMCH-IICH Messages 664
649 Summary Table for Power Management PCI Registers (PM — D31:F0) 665
650 Offset A0h: GEN_PMCON_1 – General PM Configuration 1 Register 665
651 Offset A2h: GEN_PMCON_2 – General PM Configuration 2 Register 666
652 Offset A4h: GEN_PMCON_3 – General PM Configuration 3 Register 668
653 Offset B8h: GPI_ROUT – GPI Routing Control Register 670
654 APM Register Map 670
655 Offset B2h: APM_CNT – Advanced Power Management Control Port Register 671
656 Offset B3h: APM_STS – Advanced Power Management Status Port Register 671
657 ACPI and Legacy I/O Register Map 671
658 Offset 00h: PM1_STS – Power Management 1 Status Register 673
659 Offset 02h: PM1_EN – Power Management 1 Enables Register 674
660 Offset 04h: PM1_CNT – Power Management 1 Control Register 676
661 Offset 08h: PM1_TMR – Power Management 1 Timer Register 677
662 Offset 10h: PROC_CNT – Processor Control Register 677
663 Offset 14h: LV2 – Level 2 Register 679
664 Offset 28h: GPE0_STS – General Purpose Event 0 Status Register 680
665 PMBASE Offset 2Ch: GPE0_EN – General Purpose Event 0 Enables Register 683
666 Offset 30h: SMI_EN – SMI Control and Enable Register 684
667 Offset 34h: SMI_STS – SMI Status Register 686
668 Offset 38h: ALT_GPI_SMI_EN – Alternate GPI SMI Enable Register 688
669 Offset 3Ah: ALT_GPI_SMI_STS – Alternate GPI SMI Status Register 689

670 Offset 44h: DEVTRAP_STS Register 689
671 Causes of SCI 690
672 Causes of TCO SCI 691
673 Causes of SMI# 691
674 Causes of TCO SMI# 692
675 Break Events 693
676 C0→C2→C0 Timings 696
677 Sleep State Output Conditions 697
678 Sleep Types 698
679 Causes of Wake Events 698
680 GPI Wake Events 699
681 Transitions Due To Power Failure 700
682 Transitions Due to Power Button 701
683 Transitions Due to RI# Signal 702
684 Write-Only Registers with Read Paths in Alternate Access Mode 705
685 PIC Reserved Bits Return Values 707
686 Register Write Accesses in Alternate Access Mode 707
687 TCO I/O Registers Summary Table 712
688 Offset 00 - 01h: TRLD – TCO Timer Reload and Current Value Register 713
689 Offset 02h: TDI – TCO Data In Register 713
690 Offset 03h: TDO – TCO Data Out Register 713
691 Offset 04 - 05h: TSTS1 – TCO 1 Status Register 714
692 Offset 06 - 07h: TSTS2 – TCO 2 STS Register 715
693 Offset 08 - 09h: TCTL1 – TCO 1 Control Register 717
694 Offset 0A - 0Bh: TCTL2 – TCO 2 Control Register 718
695 Offset 0C - 0Dh: TMSG – TCO MESSAGE 1 and 2 Registers 718
696 Offset 0Eh: TWDS – TCO Watchdog Status Register 719
697 Offset 10h: LE – Legacy Elimination Register 719
698 Offset 12h: TTMR – TCO_TMR Register 719
699 Event Transitions that Cause Messages 724
700 SMBus Message Format 728
701 Message Address Byte 728
702 GPIO Signal Types 730
703 GPIO Register Summary Table 731
704 Offset 00 - 03h: GPIO_USE_SEL1– GPIO Use Select 1 [31:0] Register 732
705 Offset 04 - 07h: GP_IO_SEL1 – GPIO Input/Output Select 1 [31:0] Register 732
706 Offset 0C - 0Fh: GP_LVL1 – GPIO Level 1 for Input or Output [31:0] Register 733
707 Offset 18 - 1Bh: GPO_BLINK – GPIO Blink Enable Register 734
708 Offset 2C - 2Fh: GPI_INV – GPIO Signal Invert Register 735
709 Offset 30 - 33h: GPIO_USE_SEL2 – GPIO Use Select 2 [63:32] Register 736
710 Offset 34 - 37h: GP_IO_SEL2 – GPIO Input/Output Select2 [63:32] Register 736
711 Offset 38 - 3Bh: GP_LVL2 – GPIO Level for Input or Output 2 [63:32] Register 737
712 GPIO Summary Table 738
713 USB1 Configuration Registers 740
714 Offset 00 - 03h: Identifiers Register 741
715 Offset 04 - 05h: Command Register 741
716 Offset 06 - 07h: Device Status Register 742
717 Offset 08h: Revision ID Register 743
718 Offset 09h: Programming Interface Register 743
719 Offset 0Ah: Sub Class Code Register 743
720 Offset 0Bh: Base Class Code Register 743
721 Offset 0Dh: Master Latency Timer Register 744
722 Offset 0Eh: Header Type Register 744
723 Offset 20 - 23h: Base Address Register 745

724 Offset 2C - 2Dh: USBx_SVID – USB Subsystem Vendor ID Register 745
725 Offset 2E - 2Fh: USBx_SID – USB Subsystem ID Register 746
726 Offset 3Ch: Interrupt Line Register 746
727 Offset 3Dh: Interrupt Pin Register 746
728 Offset 60h: Serial Bus Release Number Register 747
729 Offset C0 - C1h: USB Legacy Keyboard/Mouse Control Register 747
730 Offset C4h: USB Resume Enable Register 749
731 Offset C8h: USB Core Well Policy Register 749
732 Offset F8 - FBh: Manufacturer's ID Register 750
733 USB I/O Registers Summary 750
734 Offset 00 - 01h: USBCMD - USB Command Register 751
735 Run/Stop, Debug Bit Interaction SWDBG (Bit 5), Run/Stop (Bit 0) Operation 752
736 Offset 02 - 03h: USBSTS - USB Status Register 754
737 Offset 04 - 05h: USBINTR - USB Interrupt Enable Register 755
738 Offset 06 - 07h: FRNUM - Frame Number Register 755
739 Offset 08 - 0Bh: FRBASEADD - Frame List Base Address Register 756
740 Offset 0Ch: SOFMOD - Start of Frame Modify Register 756
741 Offset 10h, 12h: Port Status and Control Register 757
742 Queue Advance Criteria 764
743 USB Schedule List Traversal Decision Table 765
744 Data Field 767
745 Bits Maintained in Low Power States 771
746 USB Legacy Keyboard/Mouse Control Register Bit Implementation 771
747 SMBus signals 774
748 Function 3 Configuration Registers Summary 775
749 Offset 00 - 01h: VID – Vendor ID Register 776
750 Offset 02 - 03h: DID – Device ID Register 776
751 Offset 04 - 05h: CMD – Command Register 776
752 Offset 06 - 07h: DS – Device Status Register 777
753 Offset 08h: RID – Revision ID Register 778
754 Offset 09h: PI – Programming Interface Register 778
755 Offset 0Ah: SCC – Sub Class Code Register 778
756 Offset 0Bh: BCC – Base Class Code Register 779
757 Offset 20h - 23h: SMB_BASE – SMB Base Address Register 779
758 Offset 2C - 2Dh: SVID – SVID Register 779
759 Offset 2E - 2Fh: SID – Subsystem Identification Register 780
760 Offset 3Ch: INTLN – Interrupt Line Register 780
761 Offset 3Dh: INTPN – Interrupt Pin Register 780
762 Offset 40h: HCFG – Host Configuration Register 781
763 Offset F8 - FBh: MANID – Manufacturer's ID Register 781
764 SMB I/O Registers Summary 781
765 Offset 00h: HSTS – Host Status Register 782
766 Offset 02h: HCTL – Host Control Register 784
767 Offset 03h: HCMD – Host Command Register 786
768 Offset 04h: TSA – Transmit Slave Address Register 787
769 Offset 05h: HD0 – Data 0 Register 787
770 Offset 06h: HD1 – Data 1 Register 787
771 Offset 07h: HBD – Host Block Data Register 788
772 Offset 08h: PEC – Packet Error Check Data Register 789
773 Offset 0Ch: AUXS – Auxiliary Status Register 789
774 Offset 0Dh: AUXC – Auxiliary Control Register 790
775 Offset 0Eh: SMLC – SMLINK_PIN_CTL Register 790
776 Offset 0Fh: SMBC – SMBUS_PIN_CTL Register 791
777 Quick Protocol 792
778 Send/Receive Byte Protocol without PEC 792

779 PEC Send/Receive Order 793
780 Write Byte/Word Protocol Without PEC 793
781 PEC Bit Order 794
782 Read Byte/Word Protocol without PEC 794
783 Read Byte/Word Protocol with PEC 795
784 Process Call Protocol without PEC 796
785 Process Call Protocol with PEC 796
786 Block Read/Write Protocol without PEC 798
787 Block Read/Write Protocol with PEC 799
788 I^2C Read Command Formats 799
789 Block Write-Block Read Process Call Protocol with/without PEC 801
790 AC Timings on SM Bus 802
791 Summary of Enables for SMBALERT# 803
792 Summary of Enables for SMBus Slave Write, and SMBus Host Events 804
793 Summary of Enables for the Host Notify Command 804
794 SMB Slave Interface I/O Registers Summary 804
795 Offset 09h: RSA – Receive Slave Address Register 805
796 Offset 0Ah: SD – Slave Data Register 805
797 Offset 10h: SSTS – Slave Status Register 806
798 Offset 11h: SCMD – Slave Command Register 806
799 Offset 14h: NDA – Notify Device Address Register 807
800 Offset 16h: NDLB – Notify Data Low Byte Register 807
801 Offset 17h: NDHB – Notify Data High Byte Register 807
802 Slave Write Cycle Format 808
803 Slave Write Registers 809
804 Command Types 809
805 Slave Read Cycle Format 810
806 Data Values for Slave Read Registers 810
807 Host Notify Protocol 812
808 High Precision Event Timers Registers Summary 814
809 Offset 000 - 007h: GCAP_ID - General Capabilities and ID Register 815
810 Offset 010 - 017h: GEN_CONF - General Configuration Register 816
811 Offset 020 - 027h: GINTR_STA - General Interrupt Status Register 816
812 Offset 0F0 - 0F7h: MAIN_CNT - Main Counter Value Register 817
813 Timer *n* Configuration and Capabilities Register 818
814 Timer n Comparator Value Register 821
815 Legacy Replacement Routing 822
816 USB 1.1 and USB 2.0 Comparison 826
817 USB 2.0 Configuration Registers Summary Table 827
818 Offset 00 - 01h: VID – Vendor ID Register 828
819 Offset 02 - 03h: DID – Device Identification Register 828
820 Offset 04 - 05h: CMD – Command Register 828
821 Offset 06 - 07h: DSR – Device Status Register 830
822 Offset 08h: RID – Revision ID Register 831
823 Offset 09h: PI – Programming Interface Register 831
824 Offset 0Ah: SCC – Sub Class Code Register 832
825 Offset 0Bh: BCC – Base Class Code Register 832
826 Offset 0Dh: MLT – Master Latency Timer Register 832
827 Offset 10 - 13h: MBAR – Memory Base Address Register 833
828 Offset 2C - 2Dh: SSVID – USB 2.0 Subsystem Vendor ID Register 833
829 Offset 2E - 2Fh: SSID – USB 2.0 Subsystem ID Register 833
830 Offset 34h: CAP_PTR – Capabilities Pointer Register 834
831 Offset 3Ch: ILINE – Interrupt Line Register 834
832 Offset 3Dh: IPIN – Interrupt Pin Register 834
833 Offset 50h: PM_CID – PCI Power Management Capability ID Register 835

834 Offset 51h: PM_NEXT – Next Item Pointer #1 Register 835
835 Offset 52 - 53h: PM_CAP – Power Management Capabilities Register 836
836 Offset 54 - 55h: PM_CS – Power Management Control/Status Register 836
837 Offset 58h: DP_CID – Debug Port Capability ID Register 837
838 Offset 59h: DP_NEXT – Next Item Pointer #2 Register 838
839 Offset 5A - 5Bh: DP_BASE – Debug Port Base Offset Register 838
840 Offset 60h: SBRN – Serial Bus Release Number Register 838
841 Offset 61h: FLA – Frame Length Adjustment Register 839
842 Offset 62 - 63h: PWC – Port Wake Capability Register 839
843 Offset 68 - 6Bh: ULSEC – USB 2.0 Legacy Support Extended Capability Register 841
844 Offset 6C - 6Fh: ULSCS – USB 2.0 Legacy Support Control/Status Register 842
845 Offset 70 - 73h: ISU2SMI – Intel Specific USB 2.0 SMI Register 844
846 Offset 80h: AC – Access Control Register 845
847 Offset F8 - FBh: MANID – Manufacturer's ID Register 846
848 Host Controller Capability Registers Summary Table 847
849 Offset 00h: CAPLENGTH – Capability Length Register 847
850 Offset 02 - 03h: HCIVERSION – Host Controller Interface Version Number Register 847
851 Offset 04 - 07h: HCSPARAMS – Host Controller Structural Parameters Register 848
852 Offset 08 - 0Bh: HCCPARAMS – Host Controller Capability Parameters Register 849
853 Host Controller Operational Registers Summary Table 850
854 Offset 20 - 23h: USB 2.0CMD – USB 2.0 Command Register 851
855 Offset 24 - 27h: USB 2.0STS – USB 2.0 Status Register 853
856 Offset 28- 2Bh: USB 2.0INTR – USB 2.0 Interrupt Enable Register 855
857 Offset 2C - 2Fh: FRINDEX – Frame Index Register 856
858 Offset 30 - 33h: CTRLDSSEGMENT – Control Data Structure Segment Register 857
859 Offset 34 - 37h: PERIODICLISTBASE – Periodic Frame List Base Address Register 857
860 Offset 38 - 3Bh: ASYNCLISTADDR – Current Asynchronous List Address Register 858
861 Offset 60 - 63h: CONFIGFLAG – Configure Flag Register 858
862 Offset 64 - 67h, 68 - 6Bh, 6C - 6Fh, 70 - 73h: PORTSC – Port N Status and Control 859
863 HCRESET Bit Summary 864
864 Periodic DMA Engine Memory Reads 865
865 Asynchronous DMA Engine Reads 867
866 Asynchronous DMA Engine Writes 869
867 Host Interface Parity Errors 871
868 Effect of Resets on Port-Routing Logic 878
869 Debug Port Registers Summary Table 879
870 Offset A0 - A3h: CNTL_STS – Control/Status Register 880
871 Offset A4h: USBPID – USB PIDs Register 881
872 Offset A8 - AFh: DATABUF – Data Buffer Bytes 7:0 882
873 Offset B0h: CONFIG – Configuration Register 882
874 Debug Port Behavior 883
875 Configuration Register Summary Table 888
876 PCI Header Registers Summary Table 888
877 Offset 00 - 03h: ID – Identifiers Register 889
878 Offset 04 - 05h: CMD – Command Register 889
879 Offset 06 - 07h: STS – Device Status Register 890
880 Offset 08h: RID – Revision ID Register 891
881 Decode of PI, CC and DID Combinations 892
882 Offset 09h: PI - When Sub Class Code Register (D31:F2:Offset 0Ah, CC.SCC) = '01h' 892
883 Offset 09h: PI - When Sub Class Code Register (D31:F2:Offset 0Ah, CC.SCC) = '06h' 893
884 Offset 0A - 0Bh: CC – Class Code Register 894
885 Offset 0Dh: MLT – Master Latency Timer Register 894
886 Offset 10h: PCMDBA – Primary Command Block Base Address Register 894

887 Offset 14 - 17h: PCTLBA – Primary Control Block Base Address Register 895
888 Offset 18 - 1Bh: SCMDBA – Secondary Command Block Base Address Register 895
889 Offset 1C - 1Fh: SCTLBA – Secondary Control Block Base Address Register 896
890 Offset 20 - 23h: LBAR – Legacy Bus Master Base Address Register when SCC is SATA with AHCI PI 896
891 Offset 20 - 23h: LBAR – Legacy Bus Master Base Address Register when SCC is not SATA with AHCI PI 897
892 Offset 24 - 27h: ABAR – AHCI Base Address Register 897
893 Offset 2C - 2Fh: SS – Sub System Identifiers Register 898
894 Offset 34h: CAP – Capabilities Pointer Register 898
895 Offset 3C - 3Dh: INTR – Interrupt Information Register 898
896 Register Summary: Additional SFF-8038i Configuration Registers 899
897 Offset 40 - 41h: PTIM – Primary Timing Register 899
898 Offset 44h: D1TIM – Device 1 IDE Timing Register 901
899 Offset 48h: SYNCC – Synchronous DMA Control Register 901
900 Offset 4A - 4Bh: SYNCTIM – Synchronous DMA Timing Register 902
901 Offset 54 - 57h: IIOC – IDE I/O Configuration Register 902
902 Register Summary: PCI Power Management Capabilities Registers 903
903 Offset 70 - 71h: PID – PCI Power Management Capability ID Register 904
904 Offset 72 - 73h: PC – PCI Power Management Capabilities Register 904
905 Offset 74 - 75h: PMCS – PCI Power Management Control And Status Register 905
906 Register Summary: Message Signaled Interrupt Registers 905
907 Offset 80 - 81h: MSIID – Message Signaled Interrupt Identifiers Register 906
908 Offset 82 - 83h: MSIC – Message Signaled Interrupt Message Control Register 906
909 Offset 84 - 87h: MSIA – Message Signaled Interrupt Message Address Register 906
910 Offset 88h: MSID – Message Signaled Interrupt Message Data Register 907
911 Additional Configuration Registers Summary 907
912 Offset 90h: MAP – Port Mapping Register 908
913 Offset 91 - 93h: PCS – Port Control and Status Register 908
914 Offset A0h: SIRI – SATA Indexed Register Index 910
915 Offset A4 - A7h: STRD – SATA Indexed Register Data 911
916 Offset A8 - ABh: ACR0 – AHCI Capability Register 0 911
917 Offset AC - AFh: ACR1 – AHCI Capability Register 1 912
918 Offset C0h: ATC – APM Trapping Control Register 912
919 Offset C4h: ATS – ATM Trapping Status Register 913
920 Offset D0 - D3h: SP – Scratch Pad Register 913
921 Offset E0 - E3h: BFCS – BIST FIS Control/Status Register 914
922 Offset E4 - E7h: BFTD1 – BIST FIS Transmit Data 1 Register 916
923 Offset E8 - EBh: BFTD2 – BIST FIS Transmit Data 2 Register 916
924 Offset F8h: MID – Manufacturing ID Register 916
925 I/O Registers Summary Table 917
926 Offset 00h: PCMD – Primary Command Register 917
927 Offset 02h: PSTS – Primary Status Register 918
928 Offset 04 - 07h: PDTP – Primary Descriptor Table Pointer Register 919
929 Offset 10 - 13h: INDEX – Index Register 920
930 Offset 14 - 17h: DATA – Data Register 920
931 Memory Register Summary Table 920
932 Generic Host Controller Register Summary Table 921
933 Offset 00 - 03h: CAP – HBA Capabilities Register 921
934 Offset 04 - 07h: GHC – Global HBA Control Register 923
935 Offset 08 - 0Bh: IS – Interrupt Status Register 923
936 Offset 0C - 0Fh: PI – Ports Implemented Register 924
937 Offset 10 - 13h: VS – AHCI Version Register 925
938 Port DMA Registers for Ports[1:0] 925
939 Port DMA Registers for Ports[3:2] 926

940 Port DMA Registers for Ports[5:4] 926
941 Offset 100h, 180h, 200h, 280h, 300h, 380h:
PxCLB – Port [0-5] Command List Base Address Register 927
942 Offset 104h, 184h, 204h, 284h, 304h, 384h:
PxCLBU – Port [0-5] Command List Base Address Upper 32-bits Register 927
943 Offset 108h, 188h, 208h, 288h, 308h, 388h: PxFB – Port [0-5] FIS Base Address Register 928
944 Offset 10Ch, 18Ch, 20Ch, 28Ch, 30Ch, 38Ch:
PxFBU – Port [0-5] FIS Base Address Upper 32-bits Register 928
945 Offset 110h, 190h, 210h, 290h, 310h, 390h: PxIS –
Port [0-5] Interrupt Status Register 928
946 Offset 114h, 194h, 214h, 294h, 314h, 394h: PxIE – Port [0-5] Interrupt Enable Register . 930
947 Offset 118h, 198h, 218h, 298h, 318h, 398h: PxCMD – Port [0-5] Command Register 932
948 Port Interface Registers for Ports[1:0] 934
949 Port Interface Registers for Ports[3:2] 935
950 Port Interface Registers for Ports[5:4] 935
951 Offset 120h, 1A0h, 220h, 2A0h, 320h, 3A0h: PxTFD – Port [0-5] Task File Data Register . 936
952 Offset 124h, 1A4h, 224h, 2A4h, 324h, 3A4h: PxSIG – Port [0-5] Signature Register 937
953 Offset 128h, 1A8h, 228h, 2A8h, 328h, 3A8h:
PxSSTS – Port [0-5] Serial ATA Status Register 937
954 Offset 12Ch, 1ACh, 22Ch, 2ACh, 32Ch, 3ACh:
PxSCTL – Port [0-5] Serial ATA Control Register 938
955 Offset 130h, 1B0h, 230h, 2B0h, 330h, 3B0h:
PxSERR – Port [0-5] Serial ATA Error Register 940
956 Offset 134h, 1B4h, 234h, 2B4h, 334h, 3B4h:
PxSACT – Port [0-5] Serial ATA Active Register 943
957 Offset 138h, 1B8h, 238h, 2B8h, 338h, 3B8h:
PxCI – Port [0-5] Command Issue Register 943
958 Memory Interface Error Results 946
959 Errors During Non-DATA FIS Reception 947
960 Errors During PIO Data FIS Reception 947
961 Errors During DMA Data FIS Reception 948
962 Errors During Unknown FIS type Reception 948
963 Errors During FIS Transmission 948
964 FIS Values 949
965 MSI vs. PCI IRQ Actions 952
966 Register Summary: PCI Header Registers 959
967 Offset 00 - 03h: VID and DID – Identification Registers 959
968 Offset 04 - 05h: CMD – Device Command Register 960
969 Offset 06 - 07h: PSTS – Primary Status Register 961
970 Offset 08h: RID – Revision ID Register 962
971 Offset 09 - 0Bh: CC – Class Code Register 962
972 Offset 0Ch: CLS – Cache Line Size Register 962
973 Offset 0Dh: PLT – Primary Latency Timer Register 963
974 Offset 0Eh: HTYPE – Header Type Register 963
975 Offset 18 - 1Ah: BNUM – Bus Numbers Register 963
976 Offset 1C - 1Dh: IOBL – I/O Base and Limit Register 964
977 Offset 1E - 1Fh: SSTS – Secondary Status Register 964
978 Offset 20 - 23h: MBL – Memory Base and Limit Register 965
979 Offset 24 - 27h: PMBL – Prefetchable Memory Base and Limit Register 966
980 Offset 28 - 2Bh: PMBU32 – Prefetchable Memory Base Upper 32-Bits Register 966
981 Offset 2C - 2Fh: PMLU32 – Prefetchable Memory Limit Upper 32-Bits Register 966
982 Offset 34h: CAPP – Capabilities List Pointer Register 967
983 Offset 3C - 3Dh: INTR – Interrupt Information Register 967
984 Offset 3E - 3Fh: BCTRL – Bridge Control Register 968
985 Register Summary: Root Port Capability Structure 969

986 Offset 40 - 41h: CLIST – Capabilities List Register 969
987 Offset 42 - 43h: XCAP – PCI Express Capabilities Register 969
988 Offset 44 - 47h: DCAP – Device Capabilities Register 970
989 Offset 48 - 49h: DCTL – Device Control Register 971
990 Offset 4Ah: DSTS – Device Status Register 971
991 Offset 4C - 4Fh: LCAP – Link Capabilities Register 972
992 Offset 50 - 51h: LCTL – Link Control Register 974
993 Offset 52 - 53h: LSTS – Link Status Register 975
994 Offset 54 - 57h: SLCAP – Slot Capabilities Register 975
995 Offset 58 - 59h: SLCTL – Slot Control Register 976
996 Offset 5A - 5Bh: SLSTS – Slot Status Register 977
997 Offset 5C - 5Dh: RCTL – Root Control Register 978
998 Offset 60 - 63h: RSTS – Root Status Register 979
999 Register Summary: Message Signaled Interrupt Capability 979
1000 Offset 80 - 81h: MID – Message Signaled Interrupt Identifiers Register 979
1001 Offset 82 - 83h: MC – Message Signaled Interrupt Message Control Register 980
1002 Offset 84 - 87h: MA – Message Signaled Interrupt Message Address Register 980
1003 Offset 88 - 89h: MD – Message Signaled Interrupt Message Data Register 980
1004 Register Summary: PCI Bridge Vendor Capability 981
1005 Offset 90 - 91h: SVCAP – Subsystem Vendor Capability Register 981
1006 Offset 94 - 97h: SVID – Subsystem Vendor IDs Register 981
1007 Register Summary: PCI Power Management Capability 982
1008 Offset A0 - A1h: PMCAP – Power Management Capability Register 982
1009 Offset A2 - A3h: PMC – PCI Power Management Capabilities Register 982
1010 Offset A4 - A7h: PMCS – PCI Power Management Control And Status Register 983
1011 Register Summary: Port Configuration Capability 983
1012 Offset D8 - DBh: MPC – Miscellaneous Port Configuration Register 984
1013 Offset DC - DFh: SMSCS – SMI / SCI Status Register 985
1014 Offset F8 - FBh: MANID – Manufacturer's ID Register 986
1015 Register Summary: VC Configuration Capability 986
1016 Offset 100 - 103h: VCH – Virtual Channel Capability Header Register 987
1017 Offset 104 - 107h: VCAP1 – Virtual Channel Capability 1 Register 987
1018 Offset 108 - 10Bh: VCAP2 – Virtual Channel Capability 2 Register 987
1019 Offset 10C - 10Dh: PVC – Port Virtual Channel Control Register 988
1020 Offset 10E - 10Fh: PVS – Port Virtual Channel Status Register 988
1021 Offset 110 - 113h: V0CAP – Virtual Channel 0 Resource Capability Register 988
1022 Offset 114 - 117h: V0CTL – Virtual Channel 0 Resource Control Register 989
1023 Offset 11A - 11Bh: V0STS – Virtual Channel 0 Resource Status Register 990
1024 Register Summary: Advanced Error Reporting Capability 991
1025 Offset 140 - 143h: AECH – Advanced Error Reporting Capability Header Register 991
1026 Offset 144 - 147h: UES – Uncorrectable Error Status Register 992
1027 Offset 148 -14Bh: UEM – Uncorrectable Error Mask Register 992
1028 Offset 14C - 14Fh: UEV – Uncorrectable Error Severity Register 994
1029 Offset 150 - 153h: CES – Correctable Error Status Register 995
1030 Offset 154 - 157h: CEM – Correctable Error Mask Register 995
1031 Offset 158 - 15Bh: AECC – Advanced Error Capabilities and Control Register 996
1032 Offset 15C - 16Bh: HL – Header Log Register 996
1033 Offset 16C - 16Fh: REC – Root Error Command Register 997
1034 Offset 170 -173h: RES – Root Error Status Register 997
1035 Register Summary: Root Complex Topology Capability Structure Registers 998
1036 Offset 180 - 183h: RCTCL – Root Complex Topology Capabilities List Register 998
1037 Offset 184 - 187h: ESD – Element Self Description Register 999
1038 Offset 190 - 193h: ULD – Upstream Link Description Register 999
1039 Offset 198 - 19Fh: ULBA – Upstream Link Base Address Register 1000
1040 MSI vs. PCI IRQ Actions 1000

1041SIW Pin List 1011
1042Address Map 1012
1043Supported LPC Cycle Types 1013
1044I/O Sync Bits Description 1013
1045UART Clock Divider Support 1014
1046Baud Rate Example 1014
1047UART Signal Descriptions 1015
1048UART Register/Signal Reset States 1017
1049Internal Register Descriptions 1017
1050RBR – Receive Buffer Register 1018
1051THR – Transmit Holding Register 1018
1052IER – Interrupt Enable Register 1019
1053Interrupt Conditions 1019
1054IIR – Interrupt Identification Register 1020
1055Interrupt Identification Register Decode 1020
1056FCR – FIFO Control Register 1021
1057LCR – Line Control Register 1023
1058Line Status Register (LSR) 1025
1059MCR – Modem Control Register 1027
1060MSR – Modem Status Register 1028
1061SCR – Scratchpad Register 1029
1062DLL – Divisor Latch Register Low 1030
1063Divisor Register 1030
1064I/O Register Summary Table 1033
1065Preload Value 1 Register 0 1033
1066Preload Value 1 Register 1 1034
1067Preload Value 1 Register 2 1034
1068Preload Value 2 Register 0 1035
1069Preload Value 2 Register 1 1035
1070Preload Value 2 Register 2 1036
1071General Interrupt Status Register 1036
1072Reload Register 0 1037
1073Reload Register 1 1038
1074WDT Configuration Register 1038
1075WDT Lock Register 1039
1076SIW_SERIRQ Sampling Periods 1043
1077Configuration Register Summary 1046
1078Logical Device 4 (Serial Port 1) 1049
1079Logical Device 5 (Serial Port 2) 1050
1080Logical Device 6 (Watch Dog Timer) 1051
1081Absolute Maximum Ratings 1053
1082Operating Condition Power Supply Rails 1053
1083CMOS and SCHMITT Signals Types 1056
1084FSB Signal Group 1056
1085Processor Side-Band Signal Group 1057
1086DDR2 Signal Group 1057
1087PCI Express* Signal Group 1057
1088SMBus Signal Groups 1057
1089System Management and Power State Signal Groups 1058
1090Miscellaneous Signals and Clocks 1058
1091LPC/FWH Signal Group 1058
1092USB Signal Group 1058
1093Serial ATA Signal Group 1059
1094Interrupt Signal Group 1059
1095Watch Dog Timer and Real Time Clock Signal Group 1059

1096General Purpose I/O Signal Group 1060
1097UART Signal Group 1060
1098PCI32/33 Signal Group.....1061
1099TAP Signal Group 1061
1100FSB Interface DC Characteristics 1061
1101DDR2 Interface DC Characteristics.....1063
1102PCI Express* Differential Transmitter (TX) and Receiver (RX) DC Specifications 1064
1103PCI Express* Clock DC Characteristics.....1065
1104SMBus I/O DC Characteristics 1066
1105TAP DC Characteristics 1066
1106Miscellaneous I/O DC Characteristics.....1067
1107Processor Side-Band DC Characteristics.....1068
1108System Management and Power State DC Characteristics 1069
1109LPC/FWH DC Characteristics 1070
1110USB DC Characteristics 1071
1111Serial ATA DC Characteristics 1072
1112Interrupt's DC Characteristics 1073
1113Watch Dog Timer and Real Time Clock DC Characteristics 1073
1114General Purpose I/O DC Characteristics.....1074
1115UART DC Characteristics 1075
1116PCI32/33 DC Characteristics 1075
1117Other I/O DC Characteristics.....1076
1118Glossary Table 1113
1119Referenced Documents.....1121
1120Related Websites.....1121

Revision History

Date	Revision	Description
June 2007	004	**Modification** - Updated Section 1.2, "Supported Microprocessors" on page 49. **Reason** - Added support for Intel® Core™ 2 Duo Processor L7400, added processor data.
June 2007	003	**Modification** - Updated Section 1.2, "Supported Microprocessors" on page 49. **Reason** - Added processor speed. **Modification** - Removed 16550 compatibility from Section 1.8.15, "Serial Port" on page 62. **Reason** - Intel® 3100 Chipset is not 16550 compliant. **Modification** - Corrected THERMTRIP# Cell I/O Class in Table 76, "System Management and Power State Signals" on page 169. **Reason** - Incorrect value. Modification - Changed values in: • Figure 23, "Source in Decrement and Destination in Increment Mode Transfer (Byte Reversal)" on page 123 • Figure 24, "Source in Increment and Destination in 1-Byte Granularity Constant Mode Transfer" on page 124 • Figure 26, "Source in Increment and Destination in 4-Byte Granularity Constant Mode Transfer" on page 126 • Figure 31, "Source in Buffer Init and Destination in 1-Byte Granularity Constant Mode Transfer" on page 131 • Figure 32, "Source in Buffer Init and Destination in 2-Byte Granularity Constant Mode Transfer" on page 131 • Figure 33, "Source in Buffer Init and Destination in 4-Byte Granularity Constant Mode Transfer" on page 132 **Reason** - DCR values were incorrect. **Modification** - Changed values in: • Figure 50, "Enhanced Configuration Memory Address Map" on page 211 • Figure 58, "Dword Configuration Read Protocol" on page 232 • Figure 59, "Dword Configuration Write Protocol" on page 233 **Reason** - Incorrect values. **Modification** - Changed title and register header of Section 14.1.5.1, "Offset 3000h: TCTL – TCO Control Register" on page 544. **Reason** - This is an 8 bit register, not a 16 bit register. **Modification** - Changed NO-REBOOT bit description in Section 23.4.2, "Detecting a DOA CPU or System" on page 720. **Reason** - Clarify which bit is the NO-REBOOT bit.
June 2006	002	Updated processor support information in Chapter 1.
June 2006	001	Initial release

Glossary

Table 1. Glossary Table (Sheet 1 of 8)

Term	Definition
µBGA	Micro Ball Grid Array
ACPI	Advanced Configuration and Power Interface Specification, an industry specification of the common interfaces enabling robust operating system (OS)-directed motherboard device configuration and power management of both devices and entire systems.
AFE	Analog Front End
Agent	A logical device connected to a bus or shared interconnect that can either initiate accesses or be the target of accesses.
AHCI	Advanced Host Controller Interface, an industry specification of the interface between memory and SATA devices.
ALT Access Mode	Mode to allow the reading of write-only registers, usually used when saving/restoring register content for power management sleep state implementations.
AMC	Audio/Modem Codec
Anti-Etch	Any plane-split, void or cutout in a V_{CC} or GND plane is referred to as an anti-etch.
ASF	Alert Standards Forum
ASF	Alert Specification Format. This is the next generation of “Alert on LAN*” implementation.
Asserted	Signal is set to a level that represents logical true.
Asynchronous	1. An event that causes a change in state with no relationship to a clock signal. 2. When applied to transactions or a stream of transactions, a classification for those that do not require service within a fixed time interval.
Atomic operation	A series of two or more transactions to a device by the same initiator which are guaranteed to complete without intervening accesses by a different master. Most commonly required for a read-modify-write (RMW) operation.
BER	Bit Error Rate
BGA	Ball Grid Array
Block Locking	Ability to lock the FWH’s blocks to write-protect, read-protect, or open state.
Buffer	1. A random access memory structure. 2. The term I/O buffer is also used to describe a low-level input receiver and output driver combination.
Cx States	Processor power states (Cx states) are processor power consumption and thermal management states within the global working state, G0. • C0: Processor power state - While the processor is in this state, it executes instructions. • C1: Processor power state - This power state has the lowest latency. The hardware latency in this state must be low enough that the operating software does not consider the latency aspect of the state when deciding whether to use it. • C2: Processor power state - This state offers improved power savings over the C1 state. The worst-case hardware latency for this state is provided via the ACPI system firmware and operating software can use this information to determine when the C1 state should be used instead of the C2 state. • C3: Processor power state - This state is not supported in Intel® 3100 Chipset. The C3 state offers improved power savings over the C1 and C2 states. The worst-case hardware latency for this state is provided via the ACPI system firmware and the operating software can use this information to determine when the C2 state should be used instead of C3 state. While in the C3 state, the processor’s caches maintain state but ignore any snoops.
Cache Line	The unit of memory that is copied to and individually tracked in a cache. Specifically, 64 bytes of data or instructions aligned on a 64-byte physical address boundary.
Cfg	Used as a qualifier for transactions that target PCI configuration address space.
Character	The raw data Byte in an encoded system (i.e., the 8b value in a 8b/10b encoding scheme). This is the meaningful quantum of information to be transmitted or that is received across an encoded transmission path.

Table 1. Glossary Table (Sheet 2 of 8)

Term	Definition
CMC	Common Mode Choke
CNR	Communications and Networking Riser
Coherent	Transactions that ensure that the processor's view of memory through the cache is consistent with that obtained through the I/O subsystem.
Command	The distinct phases, cycles, or packets that make up a transaction. Requests and Completions are referred to generically as Commands.
Completion	A packet, phase, or cycle used to terminate a Transaction on a interface, or within a component. A Completion will always refer to a preceding Request and may or may not include data and/or other information.
Core Power Well	Main system power, turns off in S3 – S5
CRC	See Cyclic Redundancy Check.
Cyclic Redundancy Check	A number derived from, and stored or transmitted with, a block of data in order to detect corruption. By recalculating the CRC and comparing it to the value originally transmitted, the receiver can detect some types of transmission errors.
DDR2	DDR2 SDRAM (Double Data Rate Two Synchronous Dynamic Random Access Memory) is a system memory technology.
Deasserted	Signal is set to a level that represents logical false.
DED	Double-bit Error Detect
Deferred Transaction	A processor bus Split Transaction. The requesting agent receives a Deferred Response which allows other transactions to occur on the bus. Later, the response agent completes the original request with a separate Deferred Reply transaction.
Delayed Transaction	A transaction where the target retries an initial request, but unknown to the initiator, forwards or services the request on behalf of the initiator and stores the completion or the result of the request. The original initiator subsequently reissues the request and receives the stored completion.
DFM	Design for Manufacturability
DFT	Design for Testability
Direct Memory Access	Method of accessing memory on a system without interrupting the processors on that system.
DMA	See Direct Memory Access.

Table 1. Glossary Table (Sheet 3 of 8)

Term	Definition
DMA Channel Mode Definitions	Single A single byte (or word) is transferred. The DMA must release and re-acquire the bus for each additional byte. This is commonly-used by devices that cannot transfer the entire block of data immediately. The peripheral will request the DMA each time it is ready for another transfer. The standard PC-compatible floppy disk controller (NEC 765) only has a one-byte buffer, so it uses this mode. **Demand** Once the DMA acquires the system bus, an entire block of data is transferred, up to a maximum of 64K. If the peripheral needs additional time, it can assert the READY signal to suspend the transfer briefly. READY should not be used excessively, and for slow peripheral transfers, the Single Transfer Mode should be used instead. The difference between Block and Demand is that once a Block transfer is started, it runs until the transfer count reaches zero. DRQ only needs to be asserted until -DACK is asserted. Demand Mode will transfer one more bytes until DRQ is de-asserted, at which point the DMA suspends the transfer and releases the bus back to the processor. When DRQ is asserted later, the transfer resumes where it was suspended. Older hard disk controllers used Demand Mode until processor speeds increased to the point that it was more efficient to transfer the data using the processor, particularly if the memory locations used in the transfer were above the 16 Meg mark. **Cascade** This mechanism allows a DMA channel to request the bus, but then the attached peripheral device is responsible for placing the addressing information on the bus instead of the DMA. This is also used to implement a technique known as ``Bus Mastering''. When a DMA channel in Cascade Mode receives control of the bus, the DMA does not place addresses and I/O control signals on the bus like the DMA normally does when it is active. Instead, the DMA only asserts the -DACK signal for the active DMA channel. At this point it is up to the peripheral connected to that DMA channel to provide address and bus control signals. The peripheral has complete control over the system bus, and can do reads and/or writes to any address below 16Meg. When the peripheral is finished with the bus, it de-asserts the DRQ line, and the DMA controller can then return control to the processor or to some other DMA channel. Cascade Mode can be used to chain multiple DMA controllers together, and this is exactly what DMA Channel 4 is used for in the PC architecture. When a peripheral requests the bus on DMA channels 0, 1, 2 or 3, the slave DMA controller asserts HLDREQ, but this wire is actually connected to DRQ4 on the primary DMA controller instead of to the processor. The primary DMA controller, thinking it has work to do on Channel 4, requests the bus from the processor using HLDREQ signal. Once the processor grants the bus to the primary DMA controller, -DACK4 is asserted, and that wire is actually connected to the HLDA signal on the slave DMA controller. The slave DMA controller then transfers data for the DMA channel that requested it (0, 1, 2 or 3), or the slave DMA may grant the bus to a peripheral that wants to perform its own bus-mastering, such as a SCSI controller. Because of this wiring arrangement, only DMA channels 0, 1, 2, 3, 5, 6 and 7 are usable with peripherals on PC/AT systems. ***Note:*** DMA channel 0 was reserved for refresh operations in early IBM PC computers, but is generally available for use by peripherals in modern systems. When a peripheral is performing Bus Mastering, it is important that the peripheral transmit data to or from memory constantly while it holds the system bus. If the peripheral cannot do this, it must release the bus frequently so that the system can perform refresh operations on main memory.
Downstream	Describes commands or data flowing away from the processor-memory complex and toward I/O. The terms Upstream and Downstream are never used to describe transactions as a whole. (e.g. Downstream data may be the result of an Outbound Write, or an Inbound Read. The Completion to an Inbound Read travels Downstream.)
DW	Double Word. A legacy reference to 32 bits of data on a naturally aligned four-byte boundary (i.e. the least significant two bits of the byte address are b00). This is a legacy term used by PCI and must not be used other than in that context.
ECC	Error Correcting Code
EDMA	Enhanced DMA
EMI	Electro Magnetic Interference
EMTS	Electrical Mechanical Thermal Specification used for processor specifications.
ESD	Electrostatic Discharge

Table 1. Glossary Table (Sheet 4 of 8)

Term	Definition
EXP	High-speed Serial Interface. A generic designation for the I/O interconnect technology introduced with the Intel® 3100 Chipset IMCH, also known as PCI Express. The term EXP is used frequently throughout this document to refer to this document, and to the interface ports implementing it.
FS	Full-speed. Refers to USB.
FSB	Front Side Bus
FRU	Field Replaceable Unit
Full Duplex	A connection or channel that allows data or messages to be transmitted in opposite directions simultaneously.
FWH	Firmware Hub. A non-volatile memory device used to store the system BIOS.
Gb/s	Gigabits per second (10^9 bits per second)
GB/s	Gigabytes per second (10^9 bytes per second)
Gx States	Global system states (Gx states) apply to the entire system and are visible to the user. • G3: Mechanical off - A computer state that is entered and left by a mechanical switch. It is implied by the entry of this off state through a mechanical means that no electrical current is running through the circuitry and that it can be worked on without damaging the hardware or endangering service personnel. • G2/S5: Soft Off - A computer state where the computer consumes a minimal amount of power. • G1: Sleeping - A computer state where the computer consumes a small amount of power, user mode threads are not being executed, and the system "appears" to be off (from an end user's perspective, the display is off, and so on). • G0: Working - A computer state where the system dispatches user mode (application) threads and they execute. In this state, peripheral devices are having their power state changed dynamically.
Half Duplex	A connection or channel that allows data or messages to be transmitted in either direction, but not simultaneously.
HBA	Host Bus Adapter - necessary when connecting a peripheral to a computer that doesn't have native support for that peripheral's interface.
HCD	Host Controller Device - USB interface for programmers
HCSL	Host Clock Signal Level
HPET	High Precision Event Time (HPET) - The IA-PC HPET Architecture defines a set of timers that can be used by the operating system. The timers are defined such that the OS may be able to assign specific timers to be used directly by specific applications. Each timer can be configured to generate a separate interrupt.
HS	High-speed. Refers to USB.
I/O	1. Input/Output. 2. When used as a qualifier to a transaction type, specifies that transaction targets Intel Architecture™ specific I/O space (e.g., I/O read).
IICH	The Integrated I/O Control Hub within Intel® 3100 Chipset
IMCH	The Integrated Memory Control Hub within Intel® 3100 Chipset
Implicit Writeback	A snoop-initiated data transfer from the bus agent with the modified Cache Line to the memory controller due to an access to that line.
Inbound	A transaction where the request destination is the processor-memory complex and is sourced from I/O. The terms Inbound and Outbound refer to transactions as a whole and never to Requests or Completions in isolation. (e.g., an Inbound Read generates Downstream data, whereas an Inbound Write has Upstream data. Even more confusing, the Completion to an Inbound Read travels Downstream.)
Industry Standard Architecture	A 16-bit bus architecture associated with the IBM AT motherboard designed to connect motherboard circuitry to expansion card devices that is now considered Legacy.
Initiator	The source of requests. [IBA] An agent sending a request packet on 3GIO is referred to as the Initiator for that Transaction. The Initiator may receive a completion for the Request. [3GIO]
ISA	See Industry Standard Architecture.

Table 1. Glossary Table (Sheet 5 of 8)

Term	Definition
ISA Regime	A special legacy mode to support ISA-based devices which have been integrated into the chipset. It opens a dedicated channel from the peripheral device to the processor bus. While in this mode, the legacy device is granted exclusive accesses to memory and the ability to use Tenured Transactions.
Lane	A set of differential signal pairs, one pair for transmission and one pair for reception. A by-N Link is composed of N Lanes.
Layer	A level of abstraction commonly used in interface specifications as a tool to group elements related to a basic function of the interface within a layer and to identify key interactions between layers.
Legacy	Functional requirements handed down from previous chipsets, or PC compatibility requirements from the past.
Link	The collection of two Ports and their interconnecting Lanes. A Link is a dual simplex communications path between two components.
LPC	Low Pin Count
LPC Bus	Low Pin Count connection used to connect to the super I/O device.
LS	Low-speed. Refers to USB.
LSb	Least Significant Bit
LSB	Least Significant Byte
Master	A device or logical entity that is capable of initiating transactions. A Master is any potential Initiator.
Mbyte/s	Megabytes per second (10^6 bytes per second)
Mem	Used as a qualifier for transactions that target memory space. (For example, a Mem read to I/O.)
Metastability	A characteristic of flip flops that describes the state where the output becomes non-deterministic. Most commonly caused by a setup or hold time violation.
Multi Media Timer (MMT)	See High Precision Event Timer (HPET)
MSb	Most Significant Bit
MSB	Most Significant Byte
MSI	Message Signal Interrupt
MRL	Mechanical Retention Latch
MTBF	Mean Time Between Failures
Non-Coherent	Transactions that may cause the processor's view of memory through the cache to be different than that obtained through the I/O subsystem.
North	Usually refers to bridges. The bridge or device that is closer to the processor-memory complex.
NSI	The designation for the proprietary, internal high-speed serial interconnect between the IMCH and the IICH.
Outbound	A transaction where the request destination is I/O and is sourced from the processor-memory complex. The terms Inbound and Outbound refer to transactions as a whole and never to Requests or Completions in isolation. (For example, an Outbound Read generates Upstream data, whereas an Outbound Write has Downstream data. Even more confusing, the Completion to an Outbound Read travels Upstream.)
OWord	128 bits of data on a naturally aligned sixteen-byte boundary (e.g., the least significant four bits of the byte address are b"0000"). This is the native size of the IMCH datapath.
P2P	See Peer-to-Peer
Packet	The indivisible unit of data transfer and routing, consisting of a header, data, and CRC.
PCI	Peripheral Component Interconnect Local Bus. A 32- or 64-bit bus with multiplexed address and data lines that is primarily intended for use as an interconnect mechanism within a system between processor/memory and peripheral components or add-in cards.

Table 1. Glossary Table (Sheet 6 of 8)

Term	Definition
PCI Reset	PCIRST#. This is the secondary PCI Bus reset signal. It is a logical OR of the primary interface PLTRST# signal and the state of the Secondary Bus Reset bit of the Bridge Control register (D30:F0:Reg3Eh[6]).
PCM	Pulse Code Modulation
PEC	Packet Error Checking. This is an SMBUS 2.0 feature.
Peer-to-Peer	Transactions that occur between two devices independent of memory or the processor.
Platform Reset	IICH asserts PLTRST# to reset devices that reside on the primary PCI bus. The IICH asserts PLTRST# during power-up and when a hard reset sequence is initiated through the CF9h register. PLTRST# is driven inactive a minimum of 1 ms after both PWROK and VGATE are driven high. PLTRST# is driven for a minimum of 1 ms when initiated through the CF9h register.
Plesiochronous	From Greek, meaning almost synchronous. Describes signals that have the same nominal digital rate, but are synchronized on different clocks. Any variation in rate is constrained within specified limits, which allows a device to process the data signal without buffer underflow or overflow by making periodic compensating adjustments that repeat or delete dummy data bits. However, there is no limit to the phase difference that can accumulate between the signals over time.
Port	1. Logically, an interface between a component and a PCI Express Link. 2. Physically, a group of Transmitters and Receivers located on the same chip that define a Link.
Posted	A Transaction that is considered complete by the initiating agent or source before it actually completes at the Target of the Request or destination. All agents or devices handling the Request on behalf of the original Initiator must then treat the Transaction as being system visible from the initiating interface all the way to the final destination. Commonly refers to memory writes.
Push Model	Method of messaging or data transfer that predominately uses writes instead of reads.
Queue	A first-in first-out (FIFO) structure.
QWORD (QW)	Natural fix-sized unit of data characterized by four words where one word is 16 bits
RASUM	Reliability, Availability, Serviceability, Usability, and Manageability, which are all important characteristics of servers.
RCBA	Root Complex Base Address register at D31:F0:RegF0h. It specifies the physical address of the Intel® 3100 Chipset Configuration Space. Also used in RCBA + offset xxxxh or RCBA + xxxxh (where xxxxh is the offset) to indicate register location in the Intel® 3100 Chipset Configuration Space.
RCRB	Root Complex Register Block, as defined in the *PCI Express Specification v1.0a*. In the IICH context, it refers to a part of the Intel® 3100 Chipset Configuration Space (see RCBA, above).
Receiver	1. The Agent that receives a Packet across an interface regardless of whether it is the ultimate destination of the packet. 2. More narrowly, the circuitry required to convert incoming signals from the physical medium to more perceptible forms.
Request	A packet, phase, or cycle used to initiate a Transaction on a interface, or within a component.
Reserved	The contents or undefined states of information that are not defined at this time. Using any reserved area is not permitted. Reserved values in a register are similarly not defined and must not be used. If reserved registers or values are used, undefined behavior will result. Reserved register bits must be set to 0. However, when stated, there may be specific instances where a reserved register is either non-zero, or there may be a requirement to make it non-zero.
Resume Power Well	Trickle from power supply, only turns off when power is disconnected from wall.
Resume Reset	Signal that resets the parts of the IICH in the resume power well, generated when the trickle supply turns on.
RMW	Read-Modify-Write operation
RTC	Real-Time Clock
RTC Power Well	Powered by a coin cell battery and only turns off when the battery is drained. Powers the RTC and some resume events.

Table 1. Glossary Table (Sheet 7 of 8)

Term	Definition
RTCRESET#	Signal that resets the RTC well (but does not clear the RTC RAM memory contents).
Sx States	Sleeping states (Sx states) are types of sleeping states within the global sleeping state, G1. • S5: Soft Off state. The main memory power plane is shut down in addition to the clock synthesizer and core well power planes for the processor and Intel® 3100 Chipset. The Intel® 3100 Chipset resume well is still powered. • S4: Sleeping state - This state is only used to transition to or from the S5 state. **The S4 state is not a supported power management state in** Intel® 3100 Chipset. • S3: Suspend to RAM (STR) state - The clock synthesizer and core well power planes for the processor and Intel® 3100 Chipset are shut down, but the main memory power plane and the Intel® 3100 Chipset resume well remain active. All clocks from synthesizers are shut down during the S3 state. • S0: Awake state - Power Management state when all power planes are active.
SEC/DED	Single Error Correct/Double Error Detect - A specific data protection algorithm that distributes data and ECC across 144 bits. Enables correction of single bit errors. Allows detection of double bit errors.
SATA	Serial Advanced Technology Attachment
SATA*	Serial ATA, an industry specification of the interface for storage controllers and devices.
SEC	Single-bit Error Correct
Simplex	A connection or channel that allows data or messages to be transmitted in one direction only.
SMBus	System Management Bus. A two-wire interface through which various system components may communicate.
Snooping	A means of ensuring cache coherency by monitoring all memory accesses on a common multi-drop bus to determine if an access is to information resident within a cache.
South	Usually refers to bridges. The bridge or device that is further from the processor-memory complex.
South Port	The PCI Express downstream root port(s) on the IICH.
SPD	Serial Presence Detect
Split Lock Sequence	A sequence of transactions that occurs when the target of a lock operation is split across a processor bus data alignment or Cache Line boundary, resulting in two read transactions and two write transactions to accomplish a read-modify-write operation.
Split Transaction	A transaction that consists of distinct Request and Completion phases or packets that allow use of bus, or interconnect, by other transactions while the Target is servicing the Request.
STR	Suspend To Ram
Symbol	An expanded and encoded representation of a data Byte in an encoded system (e.g., the 10b value in a 8b/10b encoding scheme). This is the value that is transmitted over the physical medium.
Symbol Time	The amount of time required to transmit a symbol.
TAP	Test Access Port used for testability and debug of the component.
Target	A device that responds to bus Transactions. The agent receiving a request packet is referred to as the Target for that Transaction.
TCO	Total Cost of Ownership
TDM	Time Division Multiplexed
TDR	Time Domain Reflectometry
Tenured Transaction	A transaction that holds the bus or interconnect until complete, effectively blocking all other transactions while the Target is servicing the Request.
TID	See Transaction Identifier
Transaction	An overloaded term that represents an operation between two or more agents that can be comprised of multiple phases, cycles, or packets.
Transaction Identifier	A multi-bit field used to uniquely identify a transaction. Commonly used to relate a Completion with its originating Request in a Split Transaction system.

Table 1. Glossary Table (Sheet 8 of 8)

Term	Definition
Transmitter	1. The Agent that sends a Packet across an interface regardless of whether it was the original generator of the packet. 2. More narrowly, the circuitry required to drive signals onto the physical medium.
Upstream	Describes commands or data flowing toward the processor-memory complex and away from I/O. The terms Upstream and Downstream are never used to describe transactions as a whole. (For example, Upstream data may be the result of an Inbound Write, or an Outbound Read. The Completion to an Outbound Read travels Upstream.)
USB	Universal Serial Bus
VC	Virtual Channel
WDT	Watch Dog Timer
Intel® 3100 Chipset Configuration Space	The memory-mapped configuration register space whose base address is specified by the RCBA register at D31:F0:Reg F0h.
XX	Nomenclature used to describe a register's reset value when the value of the register is indeterminate. Certain registers' value at reset does not have a default value, therefore it is unknown what the value will be.

Referenced Documents

Table 2. Referenced Documents

Document Title	Order Number	Location
Advanced Configuration and Power Interface (ACPI) Specification		http://www.acpi.info/
Intel Architecture Software Developer's Manual, Volumes 1–3		http://developer.intel.com/products/processor/manuals/index.htm
Intel Corporation, Advanced Host Controller Interface Specification for Serial ATA		http://www.intel.com/technology/serialata/ahci.htm
Intel Corporation, Enhanced Host Controller Interface Specification for Universal Serial Bus		http://www.intel.com/technology/usb/spec.htm
Intel Corporation, High Precision Event Timers (HPET) Specification		http://www.intel.com/technology/architecture/hpetspec.htm
Intel® 82093AA I/O Advanced Programmable Interrupt Controller (I/O APIC)	290566	http://www.intel.com/design/chipsets/specupdt/290710.htm?iid=search&
Intel Corporation, Low Pin Count (LPC) Interface Specification		http://www.intel.com/design/chipsets/industry/lpc.htm
Intel Corporation, Multiprocessor Specification	242016	http://www.intel.com/design/archives/processors/pro/docs/242016.htm
Intel Corporation, Universal Host Controller Interface (UHCI) Specification		http://www.intel.com/technology/usb/ehcispec.htm
Intel Corporation, Universal Serial Bus (USB) Specification		http://www.intel.com/technology/usb/spec.htm
Intel Corporation, USB2 Debug Device Functional Specification		http://www.intel.com/technology/usb/download/DebugDeviceSpec_R090.pdf
Intel® Pentium® 4 FSB Specification		http://developer.intel.com/design/Pentium4/documentation.htm
JEDEC Specification		http://www.jedec.org/default.cfm
Serial ATA Specification		http://www.serialata.org/specifications.asp
SMBus Specification		http://www.smbus.org/specs/
Universal Serial Bus Specification		http://www.usb.org/developers/docs/

Table 3. Related Websites

Specification or Technology	Website
ACPI and related specifications	http://www.acpi.info/spec.htm
AT Attachment-6 with Packet Interface (ATA/ATAPI-6)	http://T13.org (T13 1410D)
BIOS boot specifications	http://www.phoenix.com/en/customer+services/white+papers-specs/
Front Panel I/O Connectivity Design Guide	http://www.formfactors.org/DeveloperResources.asp
PCI and PCI Express* related specifications	http://www.pcisig.com/specifications
Power management specifications	http://www.microsoft.com/whdc/resources/respec/specs/pmref/default.mspx

1.0 Introduction

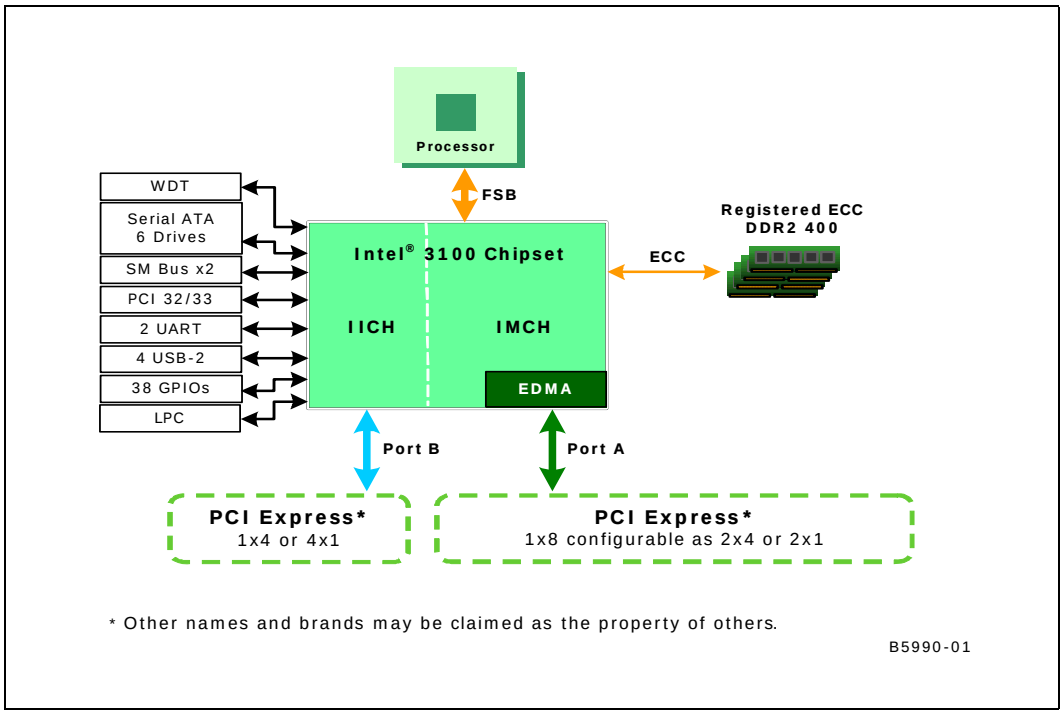
This document details the system architecture supported by the Intel® 3100 Chipset, its internal and external interfaces, and other features visible to hardware and software designers implementing an Intel® 3100 Chipset platform.

This specification includes:

- Descriptions and pin listings for all external electrical interfaces
- Descriptions of registers
- Descriptions of supported Intel® 3100 Chipset features

The Intel® 3100 Chipset is a single integrated chip that contains the functionality of a Memory Controller Hub and an I/O Controller Hub (see Figure 1). In this document the Memory Controller Hub unit and I/O Controller Hub unit in the Intel® 3100 Chipset are referenced as IMCH (Integrated Memory Controller Hub) and IICH (Integrated I/O Controller Hub) respectively. The IMCH and IICH units are connected internally through the NSI (North South Interface). The NSI is an internal bus that is not externally accessible.

Figure 1. Intel® 3100 Chipset Block Diagram



1.1 System Architecture

The Intel® 3100 Chipset provides customers an integrated system controller with an ECC memory solution in combination with high-performance, low-power processors to enable small form factor designs in the Storage, Wireless, Wire-line and Security market segments. To accomplish this, the Intel® 3100 Chipset implements numerous RASUM (Reliability, Availability, Serviceability, Usability and Manageability) features on multiple interfaces.

A Intel® 3100 Chipset system implementation consists of:

- One processor socket operating at 100 MHz (133/167 MHz with future processors)
- One Intel® 3100 Chipset
- One to four DDR2-400 DIMMs (a maximum of 4 ranks are supported)
- Bridge devices providing I/O subsystem connectivity
- Several I/O devices such as USB, SATA, etc.

The Intel® 3100 Chipset also provides one x8 PCI Express interface, which may be split into a pair of independent x4 PCI Express interfaces. Additionally, the Intel® 3100 Chipset provides one x4 PCI Express interface, which may be configured as four independent x1 PCI Express interfaces.

I/O Controller Hub (IICH) functions are integrated into the Intel® 3100 Chipset, eliminating the requirement for a legacy I/O bridge.

The Intel® 3100 Chipset also supports:

- Four USB 2.0 ports
- Six SATA ports
- One LPC bus
- Two UART port
- Two SMBus ports

1.2 Supported Microprocessors

The Intel® 3100 Chipset's FSB implementation supports a single processor configuration. Table 4 lists the processors, and frequencies supported by the Intel 3100 chipset.

Table 4. Supported Microprocessors

Processor	Freq. (GHz)	Process (nm)	L1 Cache (I+D) (KB)	L2 Cache	FSB Speed (MHz)	Streaming SIMD Extensions 2 (SSE2)	Data Prefetch Logic	Package Tech.	SpeedStep	Supports IA with Dynamic Execution
Intel® Pentium® M Processor 745	1.8	90	64	2 MB	400	YES	YES	µFC-BGA /PGA	YES	YES
Intel® Pentium® M Processor LV 738	1.4	90	64	2 MB	400	YES	YES	µFC-BGA /PGA	YES	YES
Intel® Celeron® M Processor ULV 373	1.0	90	64	512 KB	400	YES	YES	µFC-BGA	NO	YES
Intel® Celeron® M Processor 370	1.5	90	64	1 MB	400	YES	YES	µFC-BGA /PGA	NO	YES
Intel® Celeron Processor	1.66	65	64	1 MB	667	YES	YES	µFC-BGA	NO	YES
Dual-Core Intel® Xeon® Processor LV	2.0, 1.66	65	64	2 MB	667	YES	YES	µFC-BGA	YES	YES
Dual-Core Intel® Xeon® Processor ULV	1.66	65	64	2 MB	667	YES	YES	µFC-BGA	YES	YES
Intel® Celeron® M Processor ULV 423	1.067	65	64	1 MB	533	YES	YES	µFC-BGA	NO	YES
Intel® Core™ Duo processor ULV U2500	1.2	65	64	1 MB	533	YES	YES	µFC-BGA	NO	YES
Intel® Core™ 2 Duo Processor L7400	1.5	65	64	4 MB	667	YES	YES	µFC-BGA	YES	YES

1.3 Supported Memory Devices

The Intel® 3100 Chipset provides an integrated memory controller for direct-connection to one channel of DDR2-400 (unstacked) registered memory devices with ECC. Peak theoretical memory data bandwidth using DDR2-400 is 3.2 GByte/s.

A minimum memory size of 512 Mbyte is supported using a single 512 Mbit technology DIMM (9 x8 DRAMs). The maximum supported DDR2-400 memory configuration is 16 Gbyte using four 2 Gbit technology DIMMs (18 x4 DRAMs).

Table 5. DDR2-400 Memory Interface Capacities

Processor FSB Address size	512 Mbit		1 Gbit		2 Gbit	
	Min	Max	Min	Max	Min	Max
32 bit	512 MB	4096 MB	1024 MB	4096 MB	2048 MB	4096 MB
36 bit				8192 MB		16384 MB

1.4 PCI Express*

The Intel® 3100 Chipset provides one configurable x8 PCI Express interface with a maximum theoretical bandwidth of 4 GByte/s. The x8 PCI Express interface may alternatively be configured as two independent x4 PCI Express interfaces with a maximum theoretical bandwidth of 2 GBytes/s each. The Intel® 3100 Chipset also supports an additional x4 PCI Express interface with a maximum theoretical bandwidth of 2 GBytes/s which may alternatively be configured as four independent x1 PCI Express interfaces.

The Intel® 3100 Chipset is a root-class component as defined in the *PCI Express Interface Specification, Rev 1.0a*. The PCI Express interfaces support connection of the Intel® 3100 Chipset to a variety of other bridges compliant with the same revision of the *PCI Express Interface Specification, Rev 1.0a*. For example, the Intel® 82571EB Gigabit Ethernet adaptor and the Intel's PCI Express I/O processor are directly supported on any of these PCI Express ports. Other compatible PCI Express devices implement functionality such as graphics, hardware RAID controllers and TCP/IP off-load engines and PCI Express to PCI-X bridging functions. These devices are available from Intel and/or third-party vendors.

As required by the interface specification, the Intel® 3100 Chipset automatically negotiates for and train a single lane (x1) link if an attached device on any logical port fails to establish a viable x4 or x8 connection. This does not imply a capability for the Intel® 3100 Chipset to support more than two independent PCI Express ports of any width simultaneously on the x8 port, nor does it imply that the remaining three lanes of a potential x4 port are useful once the associated link has been established for x1 operation. Similarly, the Intel® 3100 Chipset will automatically negotiate for and train a single lane (x1) link if an attached device on any logical port fails to establish a viable x4 connection.

External bridge devices such as PCI or PCI-X Gigabit Ethernet or RAID storage devices are directly supported on the PCI Express ports. This does not preclude connection of the IMCH to other bridges compliant with the same revision of the *PCI Express Interface Specification*.

1.5 Supported Debug and Management Interfaces

The IMCH supports a target SMBus interface for access and control of the IMCH through its configuration registers. A Test Access Port (TAP) interface port is also supported for IMCH system debug purposes. It is capable of full read/write access to the entire internal IMCH register space.

The IICH supports XOR Chain Test Mode. This non-functional test mode is a dedicated test mode when the Intel® 3100 Chipset is not operating in its normal manner.

Platforms based on the Intel® 3100 Chipset may also make use of the Inter-Chassis Management Bus (ICMB) architecture to extend SMBus based management throughout a desegregate multi-chassis platform solution.

1.6 Supported IMCH Integrated Features

This section provides a brief overview of internal IMCH features. The list here is intended for use as an introduction and a quick reference. See Section 1.7 for more detailed descriptions of these features.

1.6.1 EDMA Controller

The IMCH includes an integrated four-channel Enhanced Direct Memory Access (EDMA) controller to perform background data transfers between locations in main memory, or from main memory to a memory-mapped I/O destination. These transfers may be individually designated to be coherent (snooped on the FSB) or non-coherent (not snooped on the FSB), providing improvements in system performance and utilization when cache coherence is managed by software rather than hardware.

Each of the four channels implements an independent set of configuration and status registers, and is capable of fully independent operation. Each channel may operate in a single block transfer mode, or a hardware traversed linked-list scatter/gather mode.

The internal EDMA controller only supports transfers between main memory locations, and transfers from a main memory source to an I/O subsystem destination. It does not support transfers between I/O interfaces, nor from an I/O interface source to a main memory destination.

1.6.2 Integrated Memory Init/Test Engine

The IMCH provides hardware-managed ECC auto-initialization and testing of all populated DRAM space under software control. Once internal configuration has been updated to reflect the types and sizes of populated DIMMs, the IMCH can traverse the populated address space issuing line-sized writes of all zero data, thereby initializing all locations with good ECC. This both greatly speeds up the mandatory memory initialization step, and frees the processor to pursue other machine initialization and configuration tasks.

Additional features have been added to the initialization engine to support high-speed population and verification of a programmable memory range with one of eight known data patterns, random data, a walking data pattern, or an explicitly specified cache line (data plus ECC). This function facilitates a limited very high-speed memory test, as well as provides a BIOS-accessible memory testing capability for potential use by management code or even by the operating system.

1.6.3 Coherent Memory Write Buffer

The IMCH includes an integrated coherent write buffer sized for 16 64-byte cache lines (a total of 1 Kbyte of storage). This feature enables the IMCH to optimize memory read latency, allowing reads to pass less critical writes en-route to the main memory store. The write buffer includes a CAM structure to enforce ordering among conflicting accesses to the same cache line, as well as to provide for read service from the write cache. In the latter case, the access to the main memory store never occurs, which both improves latency and conserves bandwidth on the memory interface.

The write buffer is capable of servicing processor read requests directly via a “hit” to the internal location containing the data without initiation of any DDR2 subsystem accesses. Inbound read requests which “hit” the write buffer result in a flush of the target data, followed by retrieval via an external read request.

1.6.4 RASUM Features

The IMCH is designed to bring enterprise level reliability, availability, serviceability, usability, and manageability to the embedded platform. All internal SRAM memory arrays are covered by parity as is the front-side bus. The PCI Express interface supports detection and automatic recovery for all transient signaling errors (99.999% available). All IMCH internal configuration register space is accessible from the system management bus (SMBus) to facilitate system management. The IMCH supports ACPI power management, PCI Express native hot-plug, and wake-from-LAN to maximize platform stand-by flexibility.

For more information on the numerous RASUM features for the memory interfaces, see the proceeding sections.

1.6.4.1 SEC-DED ECC

The IMCH supports a standard (72 bit, non-interleaved) single error correction (SEC) and double error detection (DED) ECC mechanism.

The IMCH does not provide support for operation with the ECC protection mechanism disabled. Utilization of non-ECC memory DIMMs is not supported.

1.6.4.2 Integrated Memory Scrub Engine

The IMCH includes an integrated engine to walk the populated memory space proactively seeking out soft errors in the memory subsystem. This hardware detects, logs, and corrects any single-bit ECC errors it encounters, and logs any uncorrectable errors it encounters. Both types of errors may be reported via multiple alternate mechanisms under configuration control. The scrub hardware will also execute “demand scrub” writes when correctable errors are encountered during normal operation (on demand reads, rather than scrub-initiated reads). This functionality provides incremental protection against time-based deterioration of soft memory errors from correctable to uncorrectable.

An uncorrectable error encountered by the memory scrub engine is a “speculative error.” This designation is applied because no system agent has specifically requested use of the corrupt data, and no real error condition exists in the system until that occurs. It is possible that the error resides in an unmodified page of memory that is simply dropped on a swap back to disk. If that were to occur, the speculative error would simply “vanish” from the system without any adverse consequences.

1.6.4.3 Retry on Uncorrectable Error

The IMCH includes specialized hardware to resubmit a memory read request upon detection of an uncorrectable error. When a demand fetch (as opposed to a scrub) of memory encounters an uncorrectable error as determined by the enabled ECC algorithm, the memory control hardware will cause a (single) full resubmission of the entire cache line request from memory to verify the existence of corrupt data. This feature is expected to greatly reduce or eliminate the reporting of false or transient uncorrectable errors in the DRAM array.

Any given read request will only be retried a single time on behalf of this error detection mechanism. A second occurrence of the uncorrectable error is logged and propagated according to the internal configuration settings of the IMCH. Use of this feature incurs an additional clock of latency for each and every idle system read to main memory over the default mode of operation.

1.7 IMCH Feature List

This section provides an overview of the major IMCH architectural features. Detailed usage information and operational flows, internal register bit information, and other specific details of the implementation are provided later in this document.

1.7.1 FSB Interface

- See Table 4 for the list of supported processors
- Supports single and dual-core processors
- FSB base clock rates of 100, 133, and 167 MHz[1]
  - Address and request interface double-pumped to 200, 266, and 333 MT/s, respectively
  - 64-bit data interface (+parity) quad-pumped to 400, 533, and 667 MT/s, respectively
- Up to 36-bit host interface addressing support
- Parity[2] protection on the address, data, request, and response pins
- Dynamic bus inversion to minimize power consumption on the data interface
- In-Order Queue (IOQ) depth of 12, with debug support for the one-deep non-pipelined mode
- Two outstanding DEFER cycles supported (total, not per core)
  - Only one outstanding DEFER at any time to any given I/O port
  - PCI Express memory-mapped configuration cycles “count” as an outstanding DEFER
- AGTL+ driver technology with parallel termination
- Over clocking detection support

1.7.2 Memory Interface

- Support up to four single rank or two dual rank, registered ECC DIMMs Support registered ECC DIMMs only
- DIMM parity models not supported
- Support for 15-bit address bus
  - 72 bit data bus (64-bit data plus 8-bit ECC)
- Support for base DDR2 clock rate of 200 MHz
  - Data interface double-pumped to 400 MT/s
  - Data bandwidth of 3.2 GByte/s
- Support for 512 Mbit, 1 Gbit, and 2 Gbit DRAM densities
- Support for DDR2 DIMMs using x4 or x8 DRAM technology
- Aggressive page-close policy with one-deep, look-ahead to minimize occurrence of page-miss accesses in favor of page-empty
- Support for standard SEC-DED (72, 64) ECC
- Support for automatic read retry on uncorrectable errors

1. Dependent on the processor’s system bus base frequency
2. Dependent on the processor’s support of FSB parity

- Hardware ECC auto-initialization of all populated DRAM devices under software control
  - Includes preselected hardware pattern based memory test on programmable regions

1.7.3 PCI Express* Interface in IMCH

- Supports one x8 PCI Express dual-simplex, high-speed serial I/O interface with eight striped differential pairs in each direction (outbound and inbound)
  - The interface may be unpopulated; connected to PCI, Ethernet, I/O Processor, Infiniband* bridge devices, External bridge devices (PCI or PCI-X gigabit Ethernet or RAID storage devices); or connected to any other device compliant with the same revision of the *PCI Express Specification* as the Intel® 3100 Chipset.
  - The x8 interface is capable of bifurcation into two logically independent x4 interfaces with full specification compliance at half the bandwidth capability
- This interface is referred to throughout this document as the PCI Express Port A (PEA). When configured as x8, the reference is PEA. When in x4 mode there are two available x4 ports referred to as PEA0 and PEA1.
- Raw bit-rate on the data pins of 2.5 Gbit/s, resulting in a real bandwidth per pair of 250 MByte/s given the 8/10 encoding used to transmit data across this interface
  - Maximum theoretical realized bandwidth on the x8 PCI Express interface of 2 GByte/s in each direction simultaneously, for an aggregate of 4 GByte/s.
  - Maximum theoretical realized bandwidth on the x4 PCI Express interface of 1 GByte/s in each direction simultaneously, for an aggregate of 2 GByte/s per port
- Plesiochronous operation with automatic clock extraction and phase correction at the receiver
- Hierarchical PCI-compliant configuration mechanism for downstream devices
  - Support for PCI Express memory-mapped enhanced configuration mechanism, up to 4 Kbyte per device
- 64-bit addressing support, although Intel® 3100 Chipset platforms are limited to 36-bit addressing
  - 64 bit upstream addressing (full DAC support), limited to 36 bits internally
  - 36 bit downstream addressing support, facilitating outbound DAC usage
  - Full 36 bit support for peer posted write segment accesses
- Full-speed interface self-test and diagnostic (IBIST) functionality
- Automatic discovery, negotiation, and training of PCI Express ports out of reset
  - Automatic detection of widest operational link; x8, x4 or x1
- Hardware-detected, hot-plug support at the PCI Express link level (above and beyond any PCI hot-plug support provided by a PXH or equivalent bridge)
  - Hot plug via an external SMBus connected device
  - Hot plug only supported on Device 2. I/O expander must be attached to PCI Express port to get hot plug functionality
  - Hot plug not supported on Device 3
- Run-time detection and recovery for loss of link synchronization

- 32 bit CRC (cyclic redundancy checking) on all transaction layer packets with link-level retry on error (recovery from transient errors without software-visible system failure)
- 16 bit CRC on all link message information
- ECRC (Extended CRC) support only as a end node
- Aggressive transceiver design to facilitate flexible system topologies
- Target Bit Error Rate (BER) of 10^{-12} for physical signaling interface
- Support for peer-to-peer posted writes only
- Support for coherent and non-coherent transactions through EDMA to PEA to external agent
- Support for both coherent and non-coherent traffic to memory within VC#0
  - Non-coherent implies a combination of Snoop-Not-Required and Relaxed-Ordering attributes
  - Coherent traffic implies a combination of Snoop-Required and Strong-Ordering attributes
- Support for lane reversal at all native widths, and for reversed x4 training on any x8 port
- Support for peer segment PCI interrupt forwarding to the IICH for boot from I/O
  - Legacy mode support for level-sensitive interrupt emulation without IOxAPIC support
- APIC and MSI interrupt messaging support
  - Internal score-boarding to translate messages into level-sensitive IICH pin semantics
  - XTPR based interrupt redirection for APIC messages with lowest priority tie breaking
- Support for up to 256B read completion combining
- Support for link messaging to facilitate active link and device power state management
- No support for inbound configuration or I/O traffic
- No support for inbound special cycles or writes requiring completions

1.7.4 EDMA Controller

- Four independent channels
  - Dedicated data transfer queue per channel
  - Full register set for descriptor and transfer handling per channel
- Support for transfer between main memory locations, and from memory to the I/O subsystem
- PCI Express A-segment support of traffic class to provide external prioritization of traffic. The PCI Express B-segment (Section 1.8.1) is not supported as a target for the EDMA.
- Supports transfers only between two Physical Addresses
  - Up to 36-bit (64 GB) addressing range on the Local System Memory Interface
  - Up to 36-bit addressing range on the Memory Mapped I/O Subsystem Interface
- Maximum transfer of 16 Mbyte transfers per block
- Fully programmable by the host processor

  - Configuration space mapping for EDMA engine capability and control
  - Memory-mapped space for EDMA channel-specific register sets
- Chain Mode EDMA transfer with automatic data chaining for scattering/gathering of data blocks
  - EDMA chaining continued until a “null” Descriptor Pointer is encountered
  - Support for appending a block to the end of current EDMA chain
  - Automated descriptor retrieval from local memory during chaining – single read
- Programmable independent alignment between source and destination
  - Byte aligned transfer on the Local System Memory Interface.
  - Byte aligned transfer on the I/O Subsystem Interface.
- Support for non-coherent transfers both to and from system memory on a per descriptor basis
  - Independent control of coherency for source and destination
- Programmable support for interrupt generation on block–by-block basis
  - Selectable MSI or legacy level-sensitive interrupt function
  - End of current block transfer
  - End of current chain
  - For any error causing a transfer to abort
- Increment of the source and destination address for standard transfers
- Increment of the destination and decrement of the source address to enable byte stream reversal
- Constant address mode for the destination address based on the transfer granularity to enable targeting of memory mapped I/O FIFO devices
- Buffer/Memory Initialization Mode

1.7.5 Coherent Memory Write Buffer

- Support for sixteen 64 Byte cache-lines of write data
- Fully associative conflict detection for accesses targeting memory
- Read around write support (non-conflicting) for all traffic to memory
- Read-hit support for processor traffic to memory
  - Direct data service from buffer without generation of memory traffic
- Write-hit support for memory traffic with address conflicts
  - Hardware based merging to collapse down to a single memory write
- Opportunistic and demand (buffer full) mode processing of pending writes
  - Configurable “watermark” mechanism for hardware based prioritization
  - Flush on demand via software configuration mechanism
- Parity protection on all data
- Data poisoning capability in the main store for data received with errors

1.7.6 Integrated Memory Scrub Engine

- Periodic (programmable) read-modify-write algorithm
- Support for the SEC-DED mode of operation

- Automatic correction of encountered SEC errors
- Logging of detected errors with granularity to isolate DRAM device
  - Support for logging of both first and next subsequent error
  - Count of errors beyond the first two which are logged
- Support for on-demand hardware scrub of SEC errors detected during normal operation
- Programming interface permits software suspend/resume of scrub in progress

1.7.7 Hardware Memory Initialization Engine

- Available via BIOS for hardware memory initialization and/or test
- Provides fast WHQL initialization of all populated DRAM space to “0” with good ECC
- Target region may be a single location, an entire rank, or all populated ranks
- Algorithm optimized for speed, runs at DDR2 channel saturation rate
- Test extensions permit high-speed population of a target range with a known pattern
  - Selectable hardware-generated fixed patterns: 0, 3, 5, 6, 9, A, C, F
  - Hardware generated random pattern capability
  - Explicitly stipulated data pattern including ECC
- High speed verification capability
  - Selectable write-only, verify-only, or write-read-verify per location
  - Logs error location, optional stop and escalate on error detection
- May be made available to the operating system via BIOS for security “clear to 0” function

1.7.8 System Management Functions

- Full SMBus target support
- Support for remote chassis management via the ICMB architecture
- Serial presence detect of memory devices via standard I^2C protocol (accessed via IICH)
- ACPI and PCI-PM compatible power management
  - Includes PME support comprehending PCI Express extensions
- Hot Plug support at PCI Express level
- MSI interrupt messaging and redirection support
- Hardware relay of PCI Express legacy mode PCI interrupt messages to IICH
  - Supports boot from I/O when IOxAPIC functions are unavailable

1.7.9 RASUM

- Parity protection of request and data information on the FSB
- SEC/DED ECC protection of external memory DRAM data
- Parity protection on internal data propagated through the IMCH
- CRC on data packets and hardware link-level retry on NSI to the IICH
- 32-bit CRC on data packets and hardware link-level retry on PCI Express ports

- Hardware memory initialization
  - — True “clear-to-zero” via hardware writes to all populated devices
  - — Support for hardware-based fast initialization of memory with selectable patterns
  - — Support for hardware-based fast verification of memory via accelerated scrub
- Hardware periodic memory scrubbing, including demand scrub support
- Retry on uncorrectable memory error feature
- Flexible extended error reporting capabilities
- Configurable error containment at I/O interfaces (poison/propagate or stop/escalate)
- Partial access to internal configuration registers via the TAP port
- Full access to internal configuration registers via SMBus port

1.8 IICH Feature List

This section provides a listing of architectural functionality for the major features of the IICH. Detailed usage information and operational flows, internal register bit information, and other specific details of the implementation are provided later in this document.

1.8.1 PCI Express* Interface in IICH

- Support for one x4 PCI Express dual-simplex, high-speed serial I/O interface with 4 striped differential pairs in each direction (outbound and inbound)
  - — The x4 interface is capable of bifurcation into four logically independent x1 interfaces with full specification compliance at one-fourth the bandwidth capability
- This interface is referred to throughout this document as the PCI Express Port B (PEB). When configured as x4, the reference is PEB. When in x1 mode there are four available x1 ports referred to as PEB0, PEB1, PEB2 and PEB3.
- Raw bit-rate on the data pins of 2.5 Gbit/s, resulting in a real bandwidth per pair of 250 MByte/s given the 8/10 encoding used to transmit data across this interface
  - — Maximum theoretical realized bandwidth on the x4 PCI Express interface of 1 GByte/s in each direction simultaneously, for an aggregate of 2 GByte/s.
  - — Maximum theoretical realized bandwidth on the x1 PCI Express interface of 250 MByte/s in each direction simultaneously, for an aggregate of 500 MByte/s per port
- Compliant with PCI Express 1.0a

1.8.2 Low-Pincount (LPC) Interface and Firmware Hub (FWH) Interface

- Allows connection of devices such as Super I/O, micro controllers, customer ASICs
- Supports two master/DMA devices
- Memory size up to 8 Mbit

1.8.3 Integrated Serial ATA (SATA) Host Controllers

- Independent DMA operation on six ports

  - Four ports in SATA 1.0a and AHCI mode
  - Six ports in AHCI mode only
- Data transfer rates up to 150 Mbyte/s

1.8.4 USB

- One EHCI USB 2.0 Host Controller to support a total of four ports (shared with the UHCI ports)
- Two UHCI Host Controllers to support a total of four ports (shared with the EHCI ports)
- Supports a Debug Port at USB 2.0 transfer rates
- Supports wakeup from sleeping states S3 and S5

1.8.5 Interrupt Controller

- Supports up to eight PCI Interrupt pins
- Two cascaded 82C59 with 15 interrupts
- Integrated I/O (x) APIC supporting 24 interrupts
- Serial Interrupt input for ISA legacy-compatible and PCI interrupts
- Supports PCI scheme for delivering interrupts as write cycles (rather than via PIRQ[A-H]#)
- Front-Side Message Interrupt Delivery
- Supports EOI message

1.8.6 Power Management Logic

- ACPI 2.0 Compliant
- Support for APM-based legacy power management for non-ACPI implementations
- Supports ACPI defined power states S0, S3, and, S5 (SOFF)
- ACPI Power Management Timer
- SMI# Generation
- PCI PME#
- Support for S3-Hot and S3-Cold

1.8.7 DMA Controller

- Two cascaded 8237 DMA Controllers
- Supports LPC DMA

1.8.8 Timers Based on 82C54

- System Timer, Refresh Request, Speaker Tone Output

1.8.9 High Precision Event Timers (HPET)

- Three timer comparators provided
- One-shot and periodic interrupts supported

1.8.10 Real Time Clock with 256-byte Battery-backed CMOS RAM

- Integrated components for the oscillator to reduce problems with incorrect external selections
- Lower Power DC/DC Converter implementation

1.8.11 System TCO Reduction Circuits

- Timers to detect improper processor reset and to generate SMI# and Reset upon detection of stuck processor
- Interrupt capability to OS-specific manageability extension and OS capability to call TCO BIOS
- Supports processor BIST
- Ability to disable external devices

1.8.12 SMBus

- Host interface allows processor to communicate via SMBus
- Compatible with most 2-wire components that are also I^2C compatible
- Slave interface allows internal or external microcontroller to access system resources
- SMBus 2.0 Compliant
- Flexible SMBus/SMLink architecture to optimize for ASF and eliminate board requirements for SMBus 2.0 compliance.

1.8.13 Watchdog Timer

- Selectable Prescaler:
  - Approximately 1 MHz (1 µs to 1 s)
  - Approximately 1 KHz (1 ms to 1050 s)
- 33 MHz Clock (30 ns Clock Ticks)
- Multiple Modes (WDT and Free-Running)
- Free-Running Mode:
  - One Stage Timer
  - Toggles WDT_TOUT# after programmable time
- WDT Mode:
  - Two Stage Timer (First Stage generates Interrupt, Second Stage drives WDT_TOUT# low)
  - First Stage generates an SERIRQ, NMI or SMI interrupt after Programmable time
  - Second Stage drives WDT_OUT# low or inverts the previous value
  - Used only after first timeout occurs
  - Status bit preserved in RTC well for possible error detection and correction
  - Drives WDT_TOUT# if OUTPUT is enabled
- Timer can be disabled (default state) or Locked (Hard Reset required to disable WDT)
- WDT Automatic Reload of Preload value when WDT Reload Sequence is performed

1.8.14 PCI Bus Interface

- Supports PCI Rev 2.3 specification at 33 MHz
- 120 Mbyte/sec throughput
- Supports two master devices on PCI (two request/grant pairs available)
- Support for 64-bit addressing on PCI using DAC protocol

1.8.15 Serial Port

- Two fully functional serial ports
- Configurable I/O addresses and interrupts
- 16-Byte FIFOs
- Supports up to 115 Kbps
- Programmable baud rate generator
- Modem control circuitry
- 14.7456 MHz, 33 MHz, and 48 MHz supported for UART baud clock input

1.8.16 GPIO

- 38 General Purpose I/Os; exact number varies by configuration

2.0 Configuration Register Descriptions

2.1 General Register Information

The registers found in this document follow a general formatting structure. If a register is not shown, assume it is reserved.

2.1.1 Register Attributes

Table 6 below describes the attribute acronyms.

Table 6. Configuration Table Bit Types

Type	Description
RO	Read-Only – Software/BIOS can only read this bit. Contents are either hardwired or set by hardware.
WO	Write-Only – Not supported as a bit. The write causes a hardware event to take place.
RW	Read/Write – Software/BIOS can read and write this bit.
RWC	Read/Write-Clear – Software/BIOS can read this bit and must write to a 1 to clear this bit.
RWS	Read/Write-Set – Software/BIOS can read this bit and write it to a 1. Hardware clears this bit.
RWL	Read/Write-Lock – Software/BIOS can read and write this bit. Hardware or another configuration bit can lock this bit and prevent it from being updated.
RWO	Read/Write-Once – Software/BIOS can read this bit, but can only write this bit once. It is a special form of RWL. Once any byte within a register with RWO bits has been written, the RWO bits are locked and only a reset can clear its contents. Any exceptions are clearly documented.

2.1.2 Register Nomenclature

Register values are also described here. Any value of 0 (zero) does not need a demarcation of base: binary, hexidecimal, decimal, etc. Zero can have a b or h, but it is not required. If a value is non zero, the default base is b or binary. A binary value can have b, but it is not required. H or h represents hexidecimal. If a value is hexidecimal, it will have an “h”. X means the value is undefined. Xh is XXXXb. XXh is XXXXXXXXb. And X is by default Xb.

2.2 IMCH Registers Summary

The configuration register descriptions are documented in two parts. The first part is a graphical representation of the configuration register locations in the address map. The second part is a detailed description of the registers and assigned bit definitions for the various memory-mapped registers: DDR2, NSI, and EDMA.

2.2.1 IMCH Configuration Register Maps

2.2.1.1 Device 0, Function 0 Memory Controller Hub Registers

Figure 2. Device 0, Function 0 Registers

DID		VID		00h
PCISTS		PCICMD		04h
BCC	SUBC		RID	08h
	HDR			0Ch
				10h
SMRBASE				14h
Reserved				18h
				1Ch
				20h
				24h
				28h
SID		SVID		2Ch
Reserved				30h
				34h
				38h
				3Ch
				40h
				44h
				48h
NSIBAR				4Ch
MCHCFGNS[1:0]		MCHCFG[1:0]		50h
Reserved				54h
PAM2	PAM1	PAM0	FDHC	58h
PAM6	PAM5	PAM4	PAM3	5Ch
DRB3	DRB2	DRB1	DRB0	60h
DRB7	DRB6	DRB5	DRB4	64h
Reserved for DRB expansion				68h
				6Ch
DRA3	DRA2	DRA1	DRA0	70h
Reserved for DRA expansion				74h
DRT				78h
DRC				7Ch

Reserved	DRORC	DRM		80h
ECCDIAG				84h
SDRC				88h
Reserved		CKEDIS	CKDIS	8Ch
SPARECTL				90h
DRAMISCTL				94h
DDRCSR		Reserved		98h
EXSM RAMC	SMRAM	EXSMRC	DEVPRES	9Ch
CLKGRFM0				A0h
CLKGRFM1				A4h
CLKGRMF0				A8h
CLKGRMF1				ACh
DDR2ODTC				B0h
Reserved				B4h
				B8h
				BCh
				C0h
REMAPBASE		TOLM		C4h
REMAPOFFSET		REMAPLIMIT		C8h
HECBASE		TOM		CCh
				D0h
				D4h
				D8h
SKPD				DCh
Reserved				E0h
				E4h
				E8h
				ECh
				F0h
	MCHT ST2		MCHT ST0	F4h
				F8h
				FCh

2.2.1.2 Device 0, Function 1 DRAM Controller Error Reporting Registers

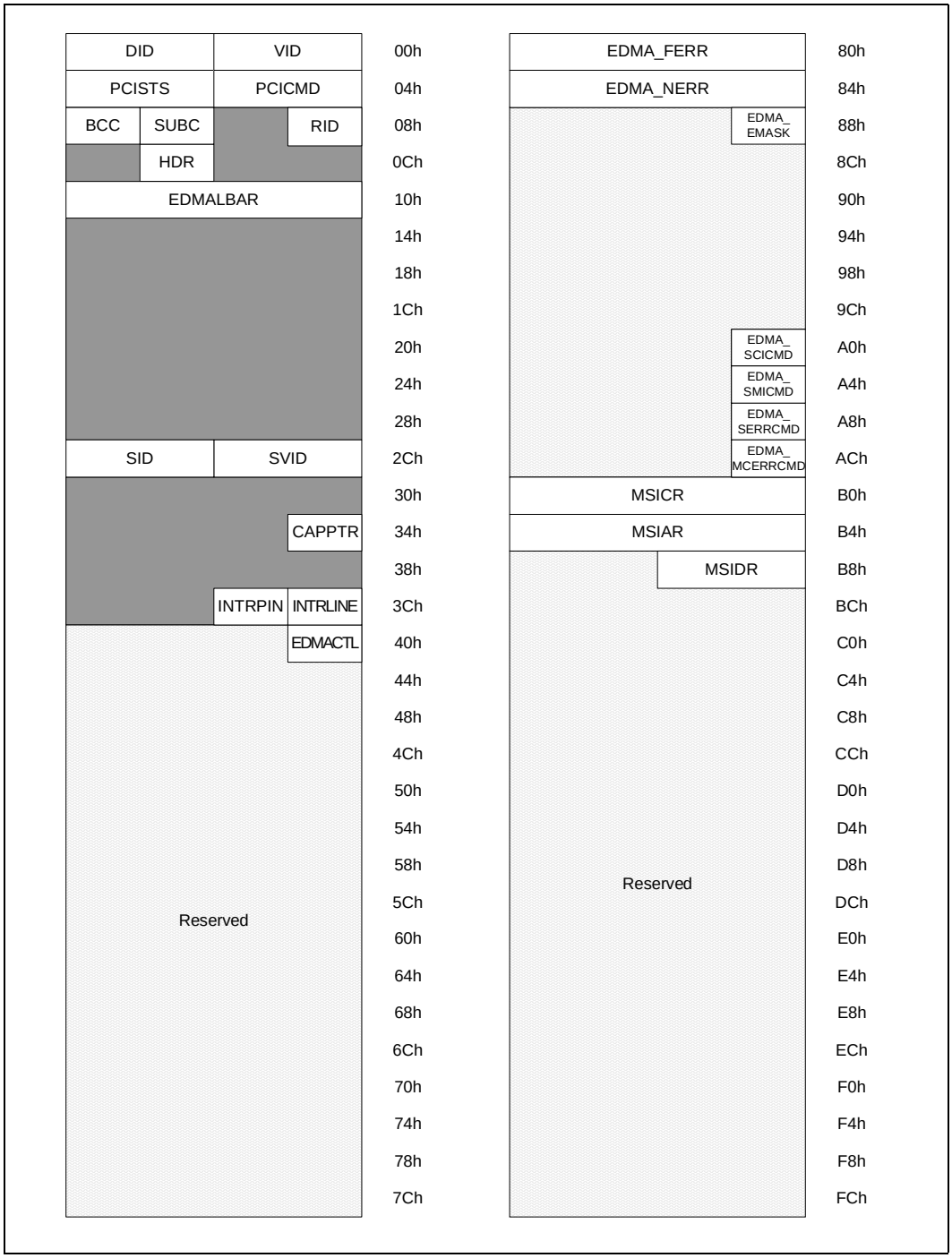
Figure 3. Device 0, Function 1 Registers

DID		VID		00h
PCISTS		PCICMD		04h
BCC	SUBC		RID	08h
	HDR	MLT		0Ch
Reserved				10h
				14h
				18h
				1Ch
				20h
				24h
				28h
SID		SVID		2Ch
				30h
				34h
				38h
				3Ch
Global_FERR				40h
Global_NERR				44h
NSI_FERR				48h
NSI_NERR				4Ch
NSI_SCICMD				50h
NSI_SMICMD				54h
NSI_SERRCMD				58h
NSI_MCERRCMD				5Ch
FSB_NERR		FSB_FERR		60h
Reserved		FSB_EMASK		64h
FSB_SMICMD		FSB_SCICMD		68h
FSB_MCERRCMD		FSB_SERRCMD		6Ch
Reserved	BUF_ NERR	Reserved	BUF_ FERR	70h
Reserved			BUF_ EMASK	74h
Reserved	BUF_ SMICMD	Reserved	BUF_ SCICMD	78h
Reserved	BUF_ MCERRCMD	Reserved	BUF_ SERRCMD	7Ch

DRAM_NERR		DRAM_FERR		80h
Reserved			DRAM_ EMASK	84h
Reserved	DRAM_ SMICMD	Reserved	DRAM_ SCICMD	88h
Reserved	DRAM_MC ERRCMD	Reserved	DRAM_ SERRCMD	8Ch
Reserved				90h
NSI_ERRSID				94h
THRESH_SEC1		THRESH_SEC0		98h
THRESH_SEC3		THRESH_SEC2		9Ch
DRAM_SECF_ADD				A0h
DRAM_DED_ADD				A4h
DRAM_SCRB_ADD				A8h
DRAM_RETR_ADD				ACh
DRAM_DED_D0A		DRAM_SEC_D0A		B0h
DRAM_DED_D1A		DRAM_SEC_D1A		B4h
DRAM_DED_D2A		DRAM_SEC_D2A		B8h
DRAM_DED_D3A		DRAM_SEC_D3A		BCh
THRESH_DED		Reserved		C0h
DRAM_SECN_ SYNDROME		DRAM_SECF_ SYNDROME		C4h
DRAM_SECN_ADD				C8h
Reserved				CCh
				D0h
				D4h
				D8h
		DIMMTHREX		DCh
HERRINJCTL				E0h
NSIERRINJCTL				E4h
BERRINJCTL				E8h
DERRINJCTL				ECh
Reserved				F0h
				F4h
				F8h
				FCh

2.2.1.3 Device 1, Function 0 EDMA Registers

Figure 4. Device 1, Function 0 EDMA Registers



2.2.1.4 Devices 2 and 3, Function 0 PEA0 and 1 Port Standard and Enhanced Registers

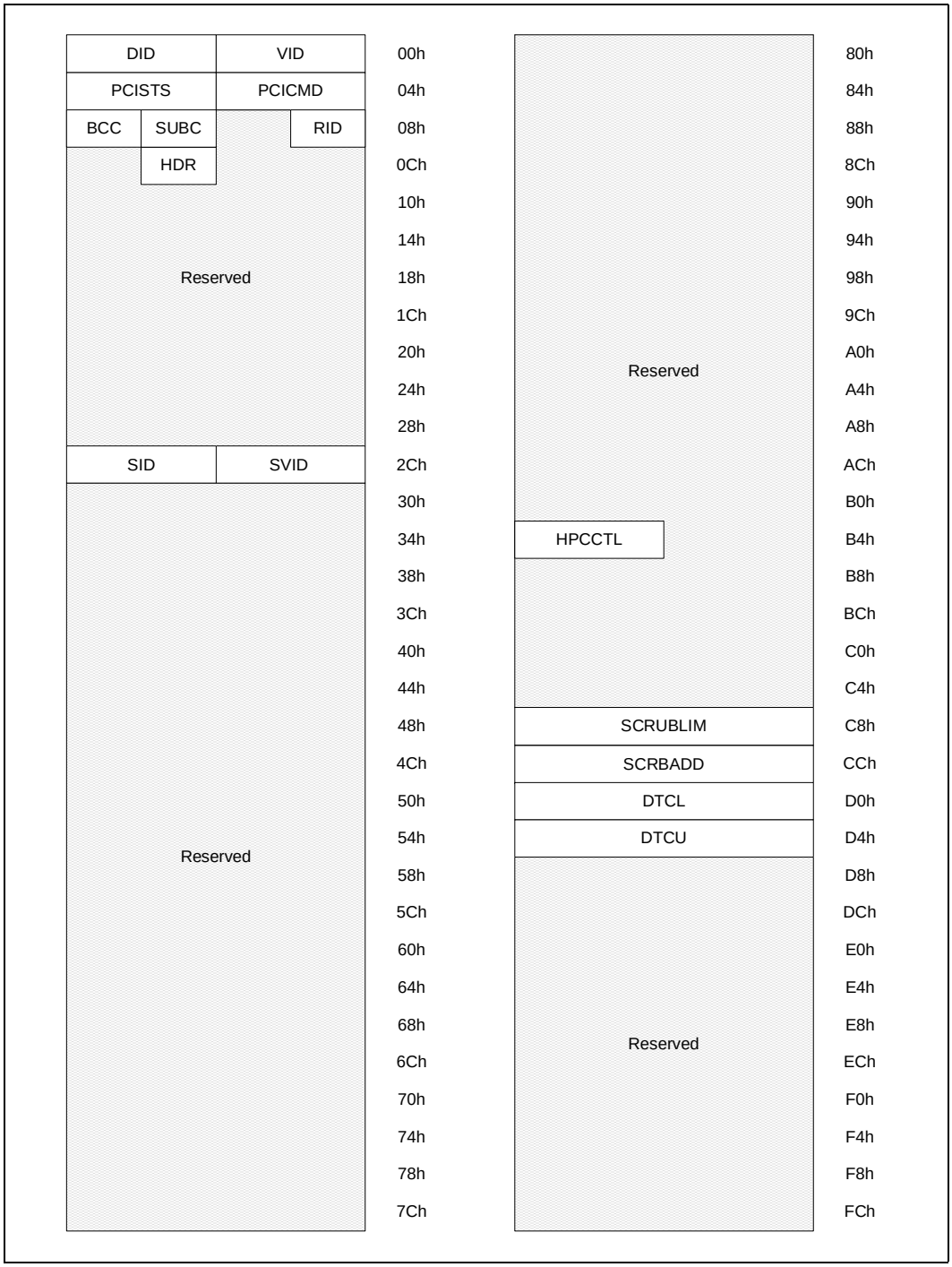
Figure 5. Devices 2 and 3, Function 0 PEA0 and 1 Port Standard and Enhanced Registers

DID		VID		00h
PCISTS		PCICMD		04h
BCC	SUBC		RID	08h
	HDR		CLS	0Ch
				10h
				14h
	SUBUSN	SBUSN	PBUSN	18h
SEC_STS		IOLIMIT	IOBASE	1Ch
MLIMIT		MBASE		20h
PMLIMIT		PMBASE		24h
			PMBASU	28h
			PMLMTU	2Ch
				30h
			CAPPTR	34h
				38h
	BCTRL	INTRPIN	INTRLINE	3Ch
				40h
VSSTS1	VSSTS0	VSCMD1	VSCMD0	44h
Reserved				48h
				4Ch
PMCAPA		PMNPTR	PMCAPID	50h
PMDATA	PMCSRBSE	PMCSR		54h
MSICAPA		MSINPTR	MSICAPID	58h
MSIAR				5Ch
Reserved		MSIDR		60h
PEACAPA		PEANPTR	PEACAPID	64h
PEADEVCAP				68h
PEADEVSTS		PEADEVCTL		6Ch
PEALNKCAP				70h
PEALNKSTS		PEALNKCTL		74h
PEASLTCAP				78h
PEASLTSTS		PEASLTCTL		7Ch

PEARPCTL	80h
PEARPSTS	84h
	88h
	8Ch
	90h
	94h
	98h
	9Ch
	A0h
	A4h
	A8h
	ACh
	B0h
	B4h
	B8h
	BCh
	C0h
Reserved	C4h
	C8h
	CCh
	D0h
	D4h
	D8h
	DCh
	E0h
	E4h
	E8h
	ECh
	F0h
	F4h
	F8h
	FCh

2.2.1.5 Device 8, Function 0 Extended Configuration Test Overflow Registers

Figure 6. Device 8, Function 0 Extended Configuration Test Overflow Registers



2.2.2 IMCH Configuration Register Summaries

This section describes the configuration-mapped registers for the Memory Controller.

2.2.2.1 Device 0, Function 0: IMCH Registers

Table 7. IMCH Controller PCI Configuration Register Map (D0, F0) (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	01h	VID	Vendor Identification Register	8086h	RO
02h	03h	DID	Device Identification Register	35B0h	RO
04h	05h	PCICMD	PCI Command Register	0006h	RO, RW
06h	07h	PCISTS	PCI Status Register	0010h	RO, RWC
08h	08h	RID	Revision Identification Register	00h	RO
0Ah	0Ah	SUBC	Sub-Class Code Register	00h	RO
0Bh	0Bh	BCC	Base Class Code Register	06h	RO
0Eh	0Eh	HDR	Header Type Register	80h	RO
14h	17h	SMRBASE	System Memory RCOMP Base Address Register	0000_0000h	RO, RW
2Ch	2Dh	SVID	Subsystem Vendor Identification Register	0000h	RWO
2Eh	2Fh	SID	Subsystem Identification Register	0000h	RWO
4Ch	4Fh	NSIBAR	Root Complex Register Block Address Register	0000_0000h	RO, RW
50h	50h	IMCHCFG0	IMCH Configuration 0 Register	04h	RO, RW
51h	51h	IMCHCFG1	IMCH Configuration 1 Register	00000h	RO, RW
52h	52h	IMCHCFGNS0	IMCH Configuration 0 Register	00h	RO, RW
53h	53h	IMCHCFGNS1	IMCH Configuration 1 Register	00h	RO, RW0C
58h	58h	FDHC	Fixed DRAM Hole Control Register	00h	RO, RW
59h	59h	PAM0	Programmable Attribute Map 0 Register	00h	RO, RW
5Ah	5Ah	PAM1	Programmable Attribute Map 1 Register	00h	RO, RW
5Bh	5Bh	PAM2	Programmable Attribute Map 2 Register	00h	RO, RW
5Ch	5Ch	PAM3	Programmable Attribute Map 3 Register	00h	RO, RW
5Dh	5Dh	PAM4	Programmable Attribute Map 4 Register	00h	RO, RW
5Eh	5Eh	PAM5	Programmable Attribute Map 5 Register	00h	RO, RW
5Fh	5Fh	PAM6	Programmable Attribute Map 6 Register	00h	RO, RW
60h	60h	DRB0	DRAM Row 0 Boundary Register	00h	RW
61h	61h	DRB1	DRAM Row 1 Boundary Register	00h	RW
62h	62h	DRB2	DRAM Row 2 Boundary Register	00h	RW
63h	63h	DRB3	DRAM Row 3 Boundary Register	00h	RW
64h	64h	DRB4	DRAM Row 4 Boundary Register	00h	RW
65h	65h	DRB5	DRAM Row 5 Boundary Register	00h	RW
66h	66h	DRB6	DRAM Row 6 Boundary Register	00h	RW
67h	67h	DRB7	DRAM Row 7 Boundary Register	00h	RW
70h	70h	DRA0	DRAM Row 0 Attribute Register	00h	RW
71h	71h	DRA1	DRAM Row 1 Attribute Register	00h	RW
72h	72h	DRA2	DRAM Row 2 Attribute Register	00h	RW
73h	73h	DRA3	DRAM Row 3 Attribute Register	00h	RW
78h	7Bh	DRT	DRAM Timing Register	859A_9604h	RW
7Ch	7Fh	DRC	DRAM Controller Mode Register	0000_0000h	RO, RW, RWO

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 7. IMCH Controller PCI Configuration Register Map (D0, F0) (Sheet 2 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
80h	81h	DRM	DRAM Mapping Register	8421h	RW
82h	82h	DRORC	Opportunistic Refresh Control Register	71h	RW
84h	87h	ECCDIAG	ECC Detection/Correction Diagnostic Register	0000_0000h	RO, RW, RWS
88h	8Bh	SDRC	DDR SDRAM Secondary Control Register	0000_0000h	RO, RW
8Ch	8Ch	CKDIS	CK/CK# Clock Disable Register	FFh	RW
8Dh	8Dh	CKEDIS	CKE/CKE# Clock Disable Register	00h	RW
90h	93h	SPARECTL	Spare Control Register	0000_0000h	RO, RW
94h	97h	DRAMISCTL	DRAM Miscellaneous Control Register	B030_0000h	RO, RW
9Ah	9Bh	DDRCSR	DDR Channel Configuration Control/Status Register	0000h	RO, RW, RWS
9Ch	9Ch	DEVPRES	Device Present Register	03h	RO, RWO
9Dh	9Dh	EXSMRC	Extended System Management RAM Control Register	00h	RO, RWL, RWC
9Eh	9Eh	SMRAM	System Management RAM Control Register	02h	RO, RW, RWC, RWS, RWL
9Fh	9Fh	EXSMRAMC	Expansion System Management RAM Control Register	07h	RO, RWC
A0h	A3h	CLKGRFM0	Clock Gearing Ratio FSB to Memory 0 Register	0015_4320h	RW
A4h	A7h	CLKGRFM1	Clock Gearing Ratio FSB to Memory 1 Register	0000_0000h	RW
A8h	ABh	CLKGRMF0	Clock Gearing Ratio Memory to FSB 0 Register	0006_5432h	RW
ACh	AFh	CLKGRMF1	Clock Gearing Ratio Memory to FSB 1 Register	0001_0000h	RW
B0h	B3h	DDR2ODTC	DDR2 ODT Control Register	0000_0000h	RW
C4h	C5h	TOLM	Top of Low Memory Register	0800h	RO, RW
C6h	C7h	REMAPBASE	Remap Base Address Register	03FFh	RO, RW
C8h	C9h	REMAPLIMIT	Remap Limit Address Register	0000h	RO, RW
CAh	CBh	REMAPOFFSET	Remap Offset Register	0000h	RO, RW
CCh	CDh	TOM	Top of Memory Register	0000h	RO, RW
CEh	CFh	HECBASE	PCI Express Port A (PEA) Enhanced Configuration Base Address Register	E000h	RO, RWO
DEh	DFh	SKPD	Scratchpad Data Register	0000h	RW
F4h	F4h	MCHTST0	IMCH Test Byte 0 Register	10h	RO, RW
F6h	F6h	MCHTST2	IMCH Test Byte 2 Register	00h	RO, RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

2.2.2.2 Device 0, Function 1: DRAM Controller Error Reporting Registers

Table 8. Error Reporting PCI Configuration Register Map (D0, F1) (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	01h	VID	Vendor Identification Register	8086h	RO
02h	03h	DID	Device Identification Register	35B1h	RO
04h	05h	PCICMD	PCI Command Register	0000h	RO, RW
06h	07h	PCISTS	PCI Status Register	0000h	RO, RWC
08h	08h	RID	Revision Identification Register	00h	RO
0Ah	0Ah	SUBC	Sub-Class Code Register	00h	RO
0Bh	0Bh	BCC	Base Class Code Register	FFh	RO
0Dh	0Dh	MLT	Master Latency Timer Register	00h	RO
0Eh	0Eh	HDR	Header Type Register	00h	RO
2Ch	2Dh	SVID	Subsystem Vendor Identification Register	0000h	RWO
2Eh	2Fh	SID	Subsystem Identification Register	0000h	RWO
40h	43h	GLOBAL_FERR	Global First Error Register	0000_0000h	RO, RWC
44h	47h	GLOBAL_NERR	Global Next Error Register	0000_0000h	RO, RWC
48h	4Bh	NSI_FERR	NSI First Error Register	0000_0000h	RO, RWC
4Ch	4Fh	NSI_NERR	NSI Next Error Register	0000_0000h	RO, RWC
50h	53h	NSI_SCICMD	NSI SCI Command Register	0000_0000h	RO, RW
54h	57h	NSI_SMICMD	NSI SMI Command Register	0000_0000h	RO, RW
58h	5Bh	NSI_SERRCMD	NSI SERR Command Register	0000_0000h	RO, RW
5Ch	5Fh	NSI_MCERRCMD	NSI MCERR Command Register	0000_0000h	RO, RW
60h	61h	FSB_FERR	FSB First Error Register	0000h	RO, RWC
62h	63h	FSB_NERR	FSB Next Error Register	0000h	RO, RWC
64h	65h	FSB_EMASK	FSB Error Mask Register	0009h	RO, RW
68h	69h	FSB_SCICMD	FSB SCI Command Register	0000h	RO, RW
6Ah	6Bh	FSB_SMICMD	FSB SMI Command Register	0000h	RO, RW
6Ch	6Dh	FSB_SERRCMD	FSB SERR Command Register	0000h	RO, RW
6Eh	6Fh	FSB_MCERRCMD	FSB MCERR Command Register	0000h	RO, RW
70h	70h	BUF_FERR	Memory Buffer First Error Register	00h	RO, RWC
72h	72h	BUF_NERR	Memory Buffer Next Error Register	00h	RO, RWC
74h	74h	BUF_EMASK	Memory Buffer Error Mask Register	00h	RO, RW
78h	78h	BUF_SCICMD	Memory Buffer SCI Command Register	00h	RO, RW
7Ah	7Ah	BUF_SMICMD	Memory Buffer SMI Command Register	00h	RO, RW
7Ch	7Ch	BUF_SERRCMD	Memory Buffer SERR Command Register	00h	RO, RW
7Eh	7Eh	BUF_MCERRCMD	Memory Buffer MCERR Command Register	00h	RO, RW
80h	81h	DRAM_FERR	DRAM First Error Register	0000h	RWC
82h	83h	DRAM_NERR	DRAM Next Error Register	0000h	RWC
84h	84h	DRAM_EMASK	DRAM Error Mask Register	00h	RW
88h	88h	DRAM_SCICMD	DRAM SCI Command Register	00h	RW
8Ah	8Ah	DRAM_SMICMD	DRAM SMI Command Register	00h	RW
8Ch	8Ch	DRAM_SERRCMD	DRAM SERR Command Register	00h	RW
8Eh	8Eh	DRAM_MCERRCMD	DRAM MCERR Command Register	00h	RW
90h	93h	NSI_EMASK	NSI Error Mask Register	0000_0000h	RO, RW
94h	97h	NSI_ERRSID	NSI Error Message Source ID Register	0000_0000h	RO

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 8. Error Reporting PCI Configuration Register Map (D0, F1) (Sheet 2 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
98h	99h	THRESH_SEC0	DIMM0 SEC Threshold Register	0000h	RW
9Ah	9Bh	THRESH_SEC1	DIMM1 SEC Threshold Register	0000h	RW
9Ch	9Dh	THRESH_SEC2	DIMM2 SEC Threshold Register	0000h	RW
9Eh	9Fh	THRESH_SEC3	DIMM3 SEC Threshold Register	0000h	RW
A0h	A3h	DRAM_SECF_ADD	DRAM First Single Bit Error Correct Address Register	0000_0000h	RO
A4h	A7h	DRAM_DED_ADD	DRAM Double Bit Error Address Register	0000_0000h	RO
A8h	ABh	DRAM_SCRB_ADD	DRAM Scrub Error Address Register	0000_0000h	RO
ACh	AFh	DRAM_RETR_ADD	DRAM DED Retry Address Register	0000_0000h	RO
B0h	B1h	DRAM_SEC_D0A	DRAM SEC Logical DIMM 0 Counter Register	0000h	RW
B2h	B3h	DRAM_DED_D0A	DRAM DED Logical DIMM 0 Counter Register	0000h	RW
B4h	B5h	DRAM_SEC_D1A	DRAM SEC Logical DIMM 1 Counter Register	0000h	RW
B6h	B7h	DRAM_DED_D1A	DRAM DED Logical DIMM 1 Counter Register	0000h	RW
B8h	B9h	DRAM_SEC_D2A	DRAM SEC Logical DIMM 2 Counter Register	0000h	RW
BAh	BBh	DRAM_DED_D2A	DRAM DED Logical DIMM 2 Counter Register	0000h	RW
BCh	BDh	DRAM_SEC_D3A	DRAM SEC Logical DIMM 3 Counter Register	0000h	RW
BEh	BFh	DRAM_DED_D3A	DRAM DED Logical DIMM 3 Counter Register	0000h	RW
C2h	C3h	THRESH_DED	Threshold for DEDs Register	0000h	RW
C4h	C5h	DRAM_SECF_ SYNDROME	DRAM First Single Error Correct Syndrome Register	0000h	RO
C6h	C7h	DRAM_SECN_ SYNDROME	DRAM Next Single Error Correct Syndrome Register	0000h	RO
C8h	CBh	DRAM_SECN_ADD	DRAM Next Single Bit Error Correct Address Register	0000_0000h	RO
DCh	DDh	DIMMTHREX	DIMM Threshold Exceeded Register	0000h	RWC
E0h	E3h	HERRCTL	Host Error Control Register	0020_0000h	RO, RW
E8h	EBh	BERRCTL	Buffer Error Control Register	0000_0000h	RO, RW
ECh	EFh	DERRCTL	DRAM Error Control Register	0000_0000h	RO, RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

2.2.2.3 Device 1, Function 0: EDMA Registers

Table 9. EDMA Configuration Register Map (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	01h	VID	Vendor Identification Register	8086h	RO
02h	03h	DID	Device Identification Register	35B5h	RO
04h	05h	PCICMD	PCI Command Register	0000h	RO, RW
06h	07h	PCISTS	PCI Status Register	0010h	RO, RWC
08h	08h	RID	Revision Identification Register	00h	RO
0Ah	0Ah	SUBC	Sub-Class Code Register	80h	RO
0Bh	0Bh	BCC	Base Class Code Register	08h	RO
0Eh	0Eh	HDR	Header Type Register	00h	RO
10h	13h	EDMALBAR	EDMA Low Base Address Register	0000_0000h	RO, RW
2Ch	2Dh	SVID	Subsystem Vendor Identification Register	0000h	RWO

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 9. EDMA Configuration Register Map (Sheet 2 of 2)

Offset Start	Offset End	Symbol	Register Name/Function	Default	Access
2Eh	2Fh	SID	Subsystem Identification Register	0000h	RWO
34h	34h	CAPPTR	Capabilities Pointer Register	B0h	RO
3Ch	3Ch	INTRLINE	Interrupt Line Register	00h	RW
3Dh	3Dh	INTRPIN	Interrupt Pin Register	01h	RO
40h	40h	EDMACTL	EDMA Control Register	08h	RO, RW
80h	83h	EDMA_FERR	EDMA First Error Register	0000_0000h	RO, RWC
84h	87h	EDMA_NERR	EDMA Next Error Register	0000_0000h	RO, RWC
88h	88h	EDMA_EMASK	EDMA Error Mask Register	00h	RW
A0h	A0h	EDMA_SCICMD	EDMA SCI Command Register	00h	RW
A4h	A4h	EDMA_SMICMD	EDMA SMI Command Register	00h	RW
A8h	A8h	EDMA_SERRCMD	EDMA SERR Command Register	00h	RW
ACh	ACh	EDMA_MCERRCMD	EDMA MCERR Command Register	00h	RW
B0h	B3h	MSICR	MSI Control Register	0002_0005h	RO, RW
B4h	B7h	MSIAR	MSI Address Register	FEE0_0000h	RW
B8h	B9h	MSIDR	MSI Data Register	0000h	RO, RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

2.2.2.4 Device 2, Function 0: PCI Express Port A Standard and Enhanced Registers

Table 10. PCI Express Port A Standard and Enhanced Configuration Register Map (Sheet 1 of 3)

Offset Start	Offset End	Symbol	Register Name/Function	Default	Access
00h	01h	VID	Vendor Identification Register	8086h	RO
02h	03h	DID	Device Identification Register	35B6h	RO
04h	05h	PCICMD	PCI Command Register	0000h	RO, RW
06h	07h	PCISTS	PCI Status Register	0010h	RO, RWC
08h	08h	RID	Revision Identification Register	00h	RO
0Ah	0Ah	SUBC	Sub-Class Code Register	04h	RO
0Bh	0Bh	BCC	Base Class Code Register	06h	RO
0Ch	0Ch	CLS	Cache Line Size Register	00h	RW
0Eh	0Eh	HDR	Header Type Register	01h	RO
18h	18h	PBUSN	Primary Bus Number Register	00h	RO
19h	19h	SBUSN	Secondary Bus Number Register	00h	RW
1Ah	1Ah	SUBUSN	Subordinate Bus Number Register	00h	RW
1Ch	1Ch	IOBASE	I/O Base Address Register	F0h	RO, RW
1Dh	1Dh	IOLIMIT	I/O Limit Address Register	00h	RW
1E	1Fh	SECSTS	Secondary Status Register	0000h	RO, RWC
20h	21h	MBASE	Memory Base Address Register	FFF0h	RO, RW
22h	23h	MLIMIT	Memory Limit Address Register	0000h	RO, RW
24h	25h	PMBASE	Prefetchable Memory Base Address Register	FFF1h	RO, RW
26h	27h	PMLIMIT	Prefetchable Memory Limit Address Register	0001h	RO, RW
28h	28h	PMBASU	Prefetchable Memory Base Upper Address Register	0Fh	RO, RW
2Ch	2Ch	PMLMTU	Prefetchable Memory Limit Upper Address Register	00h	RO, RW
34h	34h	CAPPTR	Capabilities Pointer Register	50h	RO
3Ch	3Ch	INTRLINE	Interrupt Line Register	00h	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 10. PCI Express Port A Standard and Enhanced Configuration Register Map (Sheet 2 of 3)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
3Dh	3Dh	INTRPIN	Interrupt Pin Register	01h	RWO
3Eh	3Eh	BCTRL	Bridge Control Register	00h	RO, RW
44h	44h	VSCMD0	Vendor-Specific Command Byte 0 Register	00h	RW
45h	45h	VSCMD1	Vendor-Specific Command Byte 1 Register	00h	RO, RW, RWS
46h	46h	VSSTS0	Vendor-Specific Status Byte 0 Register	00h	RO
47h	47h	VSSTS1	Vendor-Specific Status Byte 1 Register	00h	RO, RWC
50h	50h	PMCAPID	Power Management Capabilities Structure Register	01h	RO
51h	51h	PMNPTR	Power Management Next Capabilities Pointer Register	58h	RO
52h	53h	PMCAPA	Power Management Capabilities Register	C822h	RO
54h	55h	PMCSR	Power Management Status and Control Register	0000h	RO, RW
56h	56h	PMCSRBSE	Power Management Status and Control Bridge Extensions Register	00h	RO
58h	58h	MSICAPID	MSI Capabilities Structure Register	05h	RO
59h	59h	MSINPTR	MSI Next Capabilities Pointer Register	64h	RO
5Ah	5Bh	MSICAPA	MSI Capabilities Register	0002h	RO, RW
5Ch	5Fh	MSIAR	MSI Address for PCI Express Register	FEE0_0000h	RW
60h	61h	MSIDR	MSI Data Register	0000h	RW
64h	64h	PEACAPID	PCI Express Features Capabilities ID Register	10h	RO
65h	65h	PEANPTR	PCI Express Next Capabilities Pointer Register	00h	RO
66h	67h	PEACAPA	PCI Express Features Capabilities Register	0041h	RO
68h	6Bh	PEADEVCAP	PCI Express Device Capabilities Register	0000_0001h	RO
6Ch	6Dh	PEADEVCTL	PCI Express Device Control Register	0000h	RO, RW
6Eh	6Fh	PEADEVSTS	PCI Express Device Status Register	0000h	RO, RWC
70h	73h	PEALNKCAP	PCI Express Link Capabilities Register	0203_E481h	RO, RWO
74h	75h	PEALNKCTL	PCI Express Link Control Register	0000h	RO, RW, WO
76h	77h	PEALNKSTS	PCI Express Link Status Register	1001h	RO, RWO
78h	7Bh	PEASLTCAP	PCI Express Slot Capabilities Register	0000_0000h	RO, RWO
7Ch	7Dh	PEASLTCTL	PCI Express Slot Control Register	01C0h	RO, RW
7Eh	7Fh	PEASLTSTS	PCI Express Slot Status Register	0040h	RO, RWC
80h	83h	PEARPCTL	PCI Express Root Port Control Register	0000_0000h	RO, RW
84h	87h	PEARPSTS	PCI Express Root Port Status Register	0000_0000h	RO, RWC
100h	103h	ENHCAPST	Enhanced Capability Structure Register	0001_0001h	RO
104h	107h	UNCERRSTS	Uncorrectable Error Status Register	0000_0000h	RO, RWC
108h	10Bh	UNCERRMSK	Uncorrectable Error Mask Register	0000_0000h	RO, RW
10Ch	10Fh	UNCERRSEV	Uncorrectable Error Severity Register	0006_2010h	RO, RW
110h	113h	CORERRSTS	Correctable Error Status Register	0000_0000h	RO, RWC
114h	117h	CORERRMSK	Correctable Error Mask Register	0000_0000h	RO, RW
118h	11Bh	AERCACR	Advanced Error Capabilities and Control Register	000_00A0h	RO
11Ch	11Fh	HDRLOG0	Header Log DW 0 (1st 32 bits) Register	0000_0000h	RO
120h	123h	HDRLOG1	Header Log DW 1 (2nd 32 bits) Register	0000_0000h	RO
124h	127h	HDRLOG2	Header Log DW 2 (3rd 32 bits) Register	0000_0000h	RO
128h	12Bh	HDRLOG3	Header Log DW 3 (4th 32 bits) Register	0000_0000h	RO
12Ch	12Fh	RPERRCMD	Root (Port) Error Command Register	0000_0000h	RO, RW
130h	133h	RPERRMSTS	Root (Port) Error Message Status Register	0000_0000h	RO, RWC
134h	137h	ERRSID	Error Source ID Register	0000_0000h	RO
140h	143h	PEAUNITERR	PCI Express Unit Error Status Register	0000_0000h	RO, RWC
144h	147h	PEAMASKERR	PCI Express Unit Mask Error Register	0000_E000h	RO, RW
148h	14Bh	PEAERRDOCMD	PCI Express Error Do Command Register	0000_0000h	RO, RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 10. PCI Express Port A Standard and Enhanced Configuration Register Map (Sheet 3 of 3)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
14Ch	14Fh	UNCEDMASK	Uncorrectable Error Detect Mask Register	0000_0000h	RO, RW
150h	153h	COREDMASK	Correctable Error Detect Mask Register	0000_0000h	RO, RW
158h	15Bh	PEAUNITDMASK	PCI Express Unit Error Detect Mask Register	0000_0000h	RO, RW
160h	163h	PEAFERR	PCI Express First Error Register	0000_0000h	RO, RWC
164h	167h	PEANERR	PCI Express Next Error Register	0000_0000h	RO, RWC
168h	16Bh	PEAERRCTL	PCI Express Port A Error Control Register	0000_0000h	RO, RW, RWS

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

2.2.2.5 Device 3, Function 0: PCI Express* Port A1 Standard and Enhanced Registers

Table 11. Device 3, Function 0: PCI Express* Port A1 Standard and Enhanced Registers

Offset		Symbols	Register Name/Function	Default	Access
Start	End				
02h	03h	DID	Device Identification Register	*35B7h*	RO
70h	73h	EXPLNKCAP	PCI Express Link Capabilities Register	*0303_E441h*	RO, RWO
78h	7Bh	EXPSLTCAP	PCI Express Slot Capabilities Register	0000_0000h	RO, RWO

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

2.2.2.6 Device 8, Function 0: Extended Configuration Test Overflow Registers

Table 12. Extended Configuration Register Map (D8,F0)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	01h	VID	Vendor Identification Register	8086h	RO
02h	03h	DID	Device Identification Register	35C8h	RO
04h	05h	PCICMD	PCI Command Register	0000h	RO
06h	07h	PCISTS	PCI Status Register	0080h	RO
08h	08h	RID	Revision Identification Register	00h	RO
0Ah	0Ah	SUBC	Sub-Class Code Register	80h	RO
0Bh	0Bh	BCC	Base Class Code Register	08h	RO
0Eh	0Eh	HDR	Header Type Register	00h	RO
2Ch	2Dh	SVID	Subsystem Vendor Identification Register	0000h	RWO
2Eh	2Fh	SID	Subsystem Identification Register	0000h	RWO
B6h	B7h	HPCCTL	hot plug Controller Register	0014h	RO, RW
C8h	CBh	SCRUBLIM	Scrub Limit and Control Register	0000_0000h	RO, RW, RWS
CCh	CFh	SCRBADD	Scrub Address Register	0000_0000h	RO, RW
D0h	D3h	DTCL	DRAM Power Management Control Lower Register	2000_0000h	RO, RW
D4h	D7h	DTCU	DRAM Power Management Control Upper Register	0000_0000h	RO, RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

2.2.3 IMCH Memory Mapped Registers

Details for the DDR2, EDMA and NSI registers are available at this time.

2.2.3.1 Memory Mapped I/O for DDR2

This section describes the memory-mapped registers for the Memory Controller. The SMRBASE register, described in Section 13.1.1.10, "Offset 14 - 17h: SMRBASE – System Memory RCOMP Base Address Register"provides the base address for these registers. The offsets listed for the following registers are relative to this base address.

Table 13. Memory Mapped I/O for DDR2 Register Summary

Offset		Symbol	Register Name/Function	Sticky	Default	Access
Start	End					
100h	103h	DCALCSR	DCAL Control and Status Register	No	*0000_0000h*	RW
104h	107h	DCALADDR	DCAL Address Register	No	*0000_0000h*	RW
108h	10Bh	DCALDATA0	DCAL Data Register DW0	No	*0000_0000h*	RW
10Ch	10Fh	DCALDATA1	DCAL Data Register DW1	No	*0000_0000h*	RW
110h	113h	DCALDATA2	DCAL Data Register DW2	No	*0000_0000h*	RW
114h	117h	DCALDATA3	DCAL Data Register DW3	No	*0000_0000h*	RW
118h	11Bh	DCALDATA4	DCAL Data Register DW4	No	*0000_0000h*	RW
11Ch	11Fh	DCALDATA5	DCAL Data Register DW5	No	*0000_0000h*	RW
120h	123h	DCALDATA6	DCAL Data Register DW6	No	*0000_0000h*	RW
124h	127h	DCALDATA7	DCAL Data Register DW7	No	*0000_0000h*	RW
128h	12Bh	DCALDATA8	DCAL Data Register DW8	No	*0000_0000h*	RW
12Ch	12Fh	DCALDATA9	DCAL Data Register DW9	No	*0000_0000h*	RW
130h	133h	DCALDATA10	DCAL Data Register DW10	No	*0000_0000h*	RW
134h	137h	DCALDATA11	DCAL Data Register DW11	No	*0000_0000h*	RW
138h	13Bh	DCALDATA12	DCAL Data Register DW12	No	*0000_0000h*	RW
13Ch	13Fh	DCALDATA13	DCAL Data Register DW13	No	*0000_0000h*	RW
140h	143h	DCALDATA14	DCAL Data Register DW14	No	*0000_0000h*	RW
144h	147h	DCALDATA15	DCAL Data Register DW15	No	*0000_0000h*	RW
148h	14Bh	DCALDATA16	DCAL Data Register DW16	No	*0000_0000h*	RW
14Ch	14Fh	DCALDATA17	DCAL Data Register DW17	No	*0000_0000h*	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

2.2.3.2 Memory Mapped I/O for EDMA

This section describes the memory-mapped registers for the EDMA Controller. The EDMALBAR register, described in Section 13.3.1.9, "Offset 10h - 13h: EDMALBAR – EDMA Low Base Address Register" provides the base address for these registers. The offsets listed for the following registers are relative to this base address.

Table 14. Memory Mapped I/O for EDMA Register Summary (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Sticky	Default	Access
Start	End					
Memory Mapped I/O for Channel 0						
00h	**03h**	CCR0	Channel Control Register	No	0000_0000h	RW
04h	**07h**	CSR0	Channel Status Register	No	0000_0000h	RO, RWC
08h	**0Bh**	CDAR0	Current Descriptor Address Register	No	0000_0000h	RO
0Ch	**0Fh**	CDUAR0	Current Descriptor Upper Address Register	No	0000_0000h	RO
10h	**13h**	SAR0	Source Address Register	No	0000_0000h	RO
14h	**17h**	SUAR0	Source Upper Address Register	No	0000_0000h	RO
18h	**1Bh**	DAR0	Destination Address Register	No	0000_0000h	RO
1Ch	**1Fh**	DUAR0	Destination Upper Address Register	No	0000_0000h	RO
20h	**23h**	NDAR0	Next Descriptor Address Register	No	0000_0000h	RWL

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 14. Memory Mapped I/O for EDMA Register Summary (Sheet 2 of 2)

Offset		Symbol	Register Name/Function	Sticky	Default	Access
Start	End					
24h	27h	NDUAR0	Next Descriptor Upper Address Register	No	0000_0000h	RWL
28h	2Bh	TCR0	Transfer Count Register	No	0000_0000h	RO
2Ch	2Fh	DCR0	Descriptor Control Register	No	0000_0000h	RO
Memory Mapped I/O for Channel 1						
40h	43h	CCR1	Channel Control Register	No	0000_0000h	RO, RW, RWS
44h	47h	CSR1	Channel Status Register	No	0000_0000h	RO, RWC
48h	4Bh	CDAR1	Current Descriptor Address Register	No	0000_0000h	RO
4Ch	4Fh	CDUAR1	Current Descriptor Upper Address Register	No	0000_0000h	RO
50h	53h	SAR1	Source Address Register	No	0000_0000h	RO
54h	57h	SUAR1	Source Upper Address Register	No	0000_0000h	RO
58h	5Bh	DAR1	Destination Address Register	No	0000_0000h	RO
5Ch	5Fh	DUAR1	Destination Upper Address Register	No	0000_0000h	RO
60h	63h	NDAR1	Next Descriptor Address Register	No	0000_0000h	RWL
64h	67h	NDUAR1	Next Descriptor Upper Address Register	No	0000_0000h	RWL
68 h	6Bh	TCR1	Transfer Count Register	No	0000_0000h	RO
6Ch	6Fh	DCR1	Descriptor Control Register	No	0000_0000h	RO
Memory Mapped I/O for Channel 2						
80h	83h	CCR2	Channel Control Register	No	0000_0000h	RO, RW, RWS
84h	87h	CSR2	Channel Status Register	No	0000_0000h	RO, RWC
88h	8Bh	CDAR2	Current Descriptor Address Register	No	0000_0000h	RO
8Ch	8Fh	CDUAR2	Current Descriptor Upper Address Register	No	0000_0000h	RO
90h	93h	SAR2	Source Address Register	No	0000_0000h	RO
94h	97h	SUAR2	Source Upper Address Register	No	0000_0000h	RO
98h	9Bh	DAR2	Destination Address Register	No	0000_0000h	RO
9Ch	9Fh	DUAR2	Destination Upper Address Register	No	0000_0000h	RO
A0h	A3h	NDAR2	Next Descriptor Address Register	No	0000_0000h	RWL
A4h	A7h	NDUAR2	Next Descriptor Upper Address Register	No	0000_0000h	RWL
A8h	ABh	TCR2	Transfer Count Register	No	0000_0000h	RO
ACh	AFh	DCR2	Descriptor Control Register	No	0000_0000h	RO
Memory Mapped I/O for Channel 3						
C0h	C3h	CCR3	Channel Control Register	No	0000_0000h	RO, RW, RWS
C4h	C7h	CSR3	Channel Status Register	No	0000_0000h	RO, RWC
C8h	CBh	CDAR3	Current Descriptor Address Register	No	0000_0000h	RO
CCh	CFh	CDUAR3	Current Descriptor Upper Address Register	No	0000_0000h	RO
D0h	D3h	SAR3	Source Address Register	No	0000_0000h	RO
D4h	D7h	SUAR3	Source Upper Address Register	No	0000_0000h	RO
D8h	DBh	DAR3	Destination Address Register	No	0000_0000h	RO
DCh	DFh	DUAR3	Destination Upper Address Register	No	0000_0000h	RO
E0h	E3h	NDAR3	Next Descriptor Address Register	No	0000_0000h	RWL
E4h	E7h	NDUAR3	Next Descriptor Upper Address Register	No	0000_0000h	RWL
E8h	EBh	TCR3	Transfer Count Register	No	0000_0000h	RO
ECh	EFh	DCR3	Descriptor Control Register	No	0000_0000h	RO
Memory Mapped I/O for the EDMA Controller						
100h	103h	DCGC	EDMA Controller Global Command Register	No	0000_0000h	RO, RW
104h	107h	DCGS	EDMA Controller Global Status Register	No	0000_0000h	RO

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

2.2.3.3 Memory Mapped I/O for NSI

This section describes the memory-mapped registers for the North South Interface (NSI). The offsets listed for the following registers are relative to this base address.

Table 15. Memory Mapped I/O Registers for NSI Register Summary

Offset		Symbol	Register Name/Function	Sticky	Default	Access
Start	End					
00h	03h	NSIVCECH	NSI Virtual Channel Enhanced Capability Header Register	No	0401_0002h	RO
04h	07h	NSIPVCCAP1	NSI Port VC Capability Register 1	No	0000_0000h	RO
08h	0Bh	NSIPVCCAP2	NSI Port VC Capability Register 2	No	0000_0001h	RO
0Ch	0Dh	NSIPVCCTL	NSI Port VC Control Register	No	0000h	RO, RW
10h	13h	NSIVC0RCAP	NSI VC0 Resource Capability Register	No	0000_0001h	RO
14h	17h	NSIVC0RCTL	NSI VC0 Resource Control Register	No	8000_00FFh	RO, RW
1Ah	1Bh	NSIVC0RSTS	NSI VC0 Resource Status Register	No	0002h	RW
80h	83h	NSIRCILCECH	NSI Root Complex Internal Link Control Enhanced Capability Header Register	No	0001_0006h	RO
84h	87h	NSILCAP	NSI Link Capabilities Register	No	0003_A041h	RO, RWO

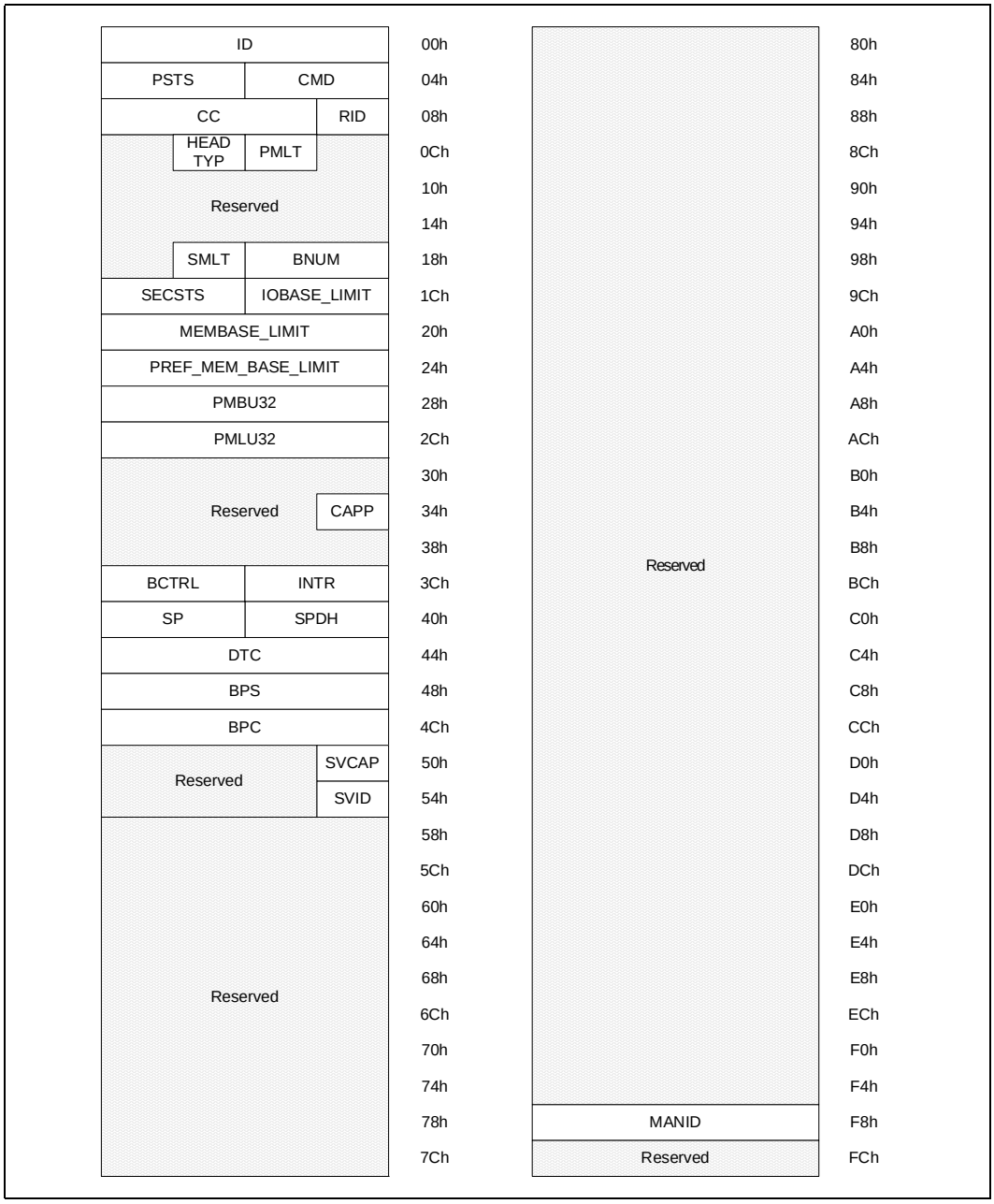
Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

2.3 IICH Registers Summary

2.3.1 IICH Configuration Register Maps

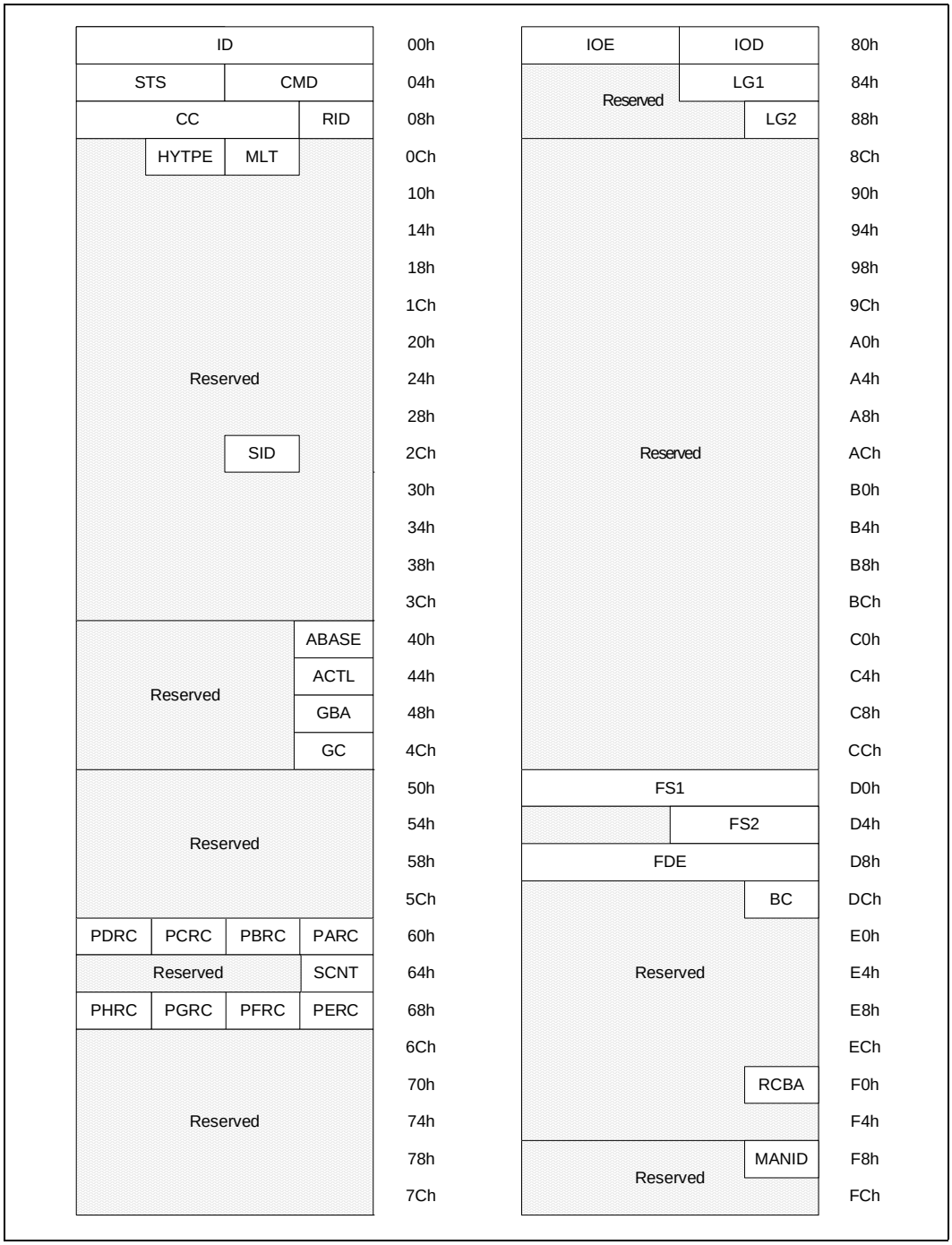
2.3.1.1 PCI to PCI Bridge

Figure 7. PCI to PCI Bridge Registers



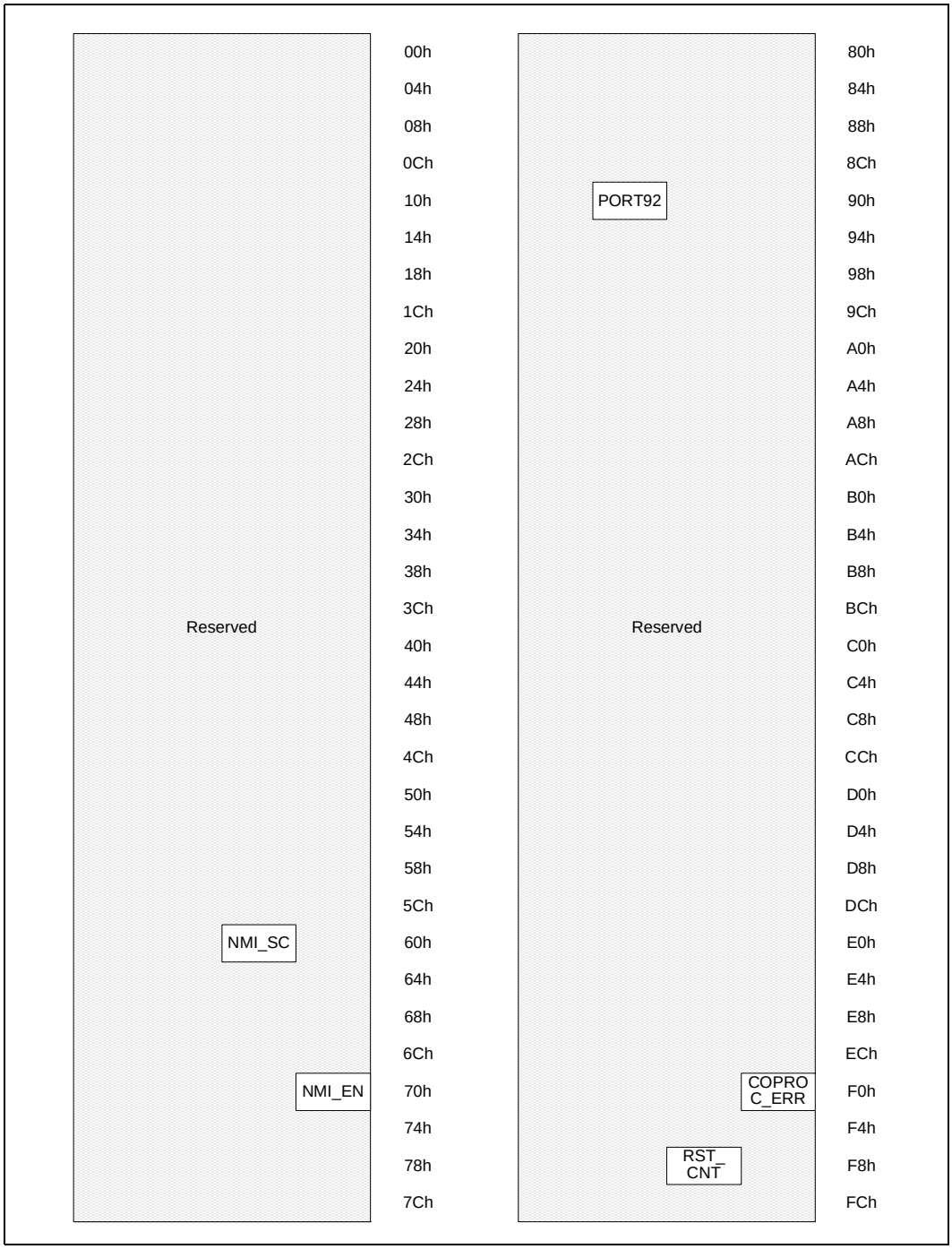
2.3.1.2 LPC I/F Device 31, Function 0

Figure 8. LPC I/F Device 31, Function 0 Registers



2.3.1.3 CPU I/F

Figure 9. CPU I/F Registers



2.3.1.4 TCU I/O

Figure 10. TCO I/O Registers

TDO	TDI	TRLD		00h		80h
TSTS2		TSTS1		04h		84h
TCTL2		TCTL1		08h		88h
RSV	TWDS	TMSG		0Ch		8Ch
	TTMR		LE	10h		90h
				14h		94h
				18h		98h
				1Ch		9Ch
				20h		A0h
				24h		A4h
				28h		A8h
				2Ch		ACh
				30h		B0h
				34h		B4h
				38h		B8h
				3Ch		BCh
Reserved				40h	Reserved	C0h
				44h		C4h
				48h		C8h
				4Ch		CCh
				50h		D0h
				54h		D4h
				58h		D8h
				5Ch		DCh
				60h		E0h
				64h		E4h
				68h		E8h
				6Ch		ECh
				70h		F0h
				74h		F4h
				78h		F8h
				7Ch		FCh

2.3.1.5 GPIO

Figure 11. GPIO Registers

GPIO_USE_SEL	00h	Reserved	80h
GP_IO_SEL	04h		84h
Reserved	08h		88h
GP_LVL	0Ch		8Ch
Reserved	10h		90h
	14h		94h
GPO_BLINK	18h		98h
Reserved	1Ch		9Ch
	20h		A0h
	24h		A4h
	28h		A8h
GPI_INV	2Ch		ACh
GPIO_USE_SEL2	30h		B0h
GP_IO_SEL2	34h		B4h
GP_LVL2	38h		B8h
Reserved	3Ch		BCh
	40h		C0h
	44h		C4h
	48h		C8h
	4Ch		CCh
	50h		D0h
	54h		D4h
	58h		D8h
	5Ch		DCh
	60h		E0h
	64h		E4h
	68h		E8h
	6Ch		ECh
	70h		F0h
	74h		F4h
	78h		F8h
	7Ch		FCh

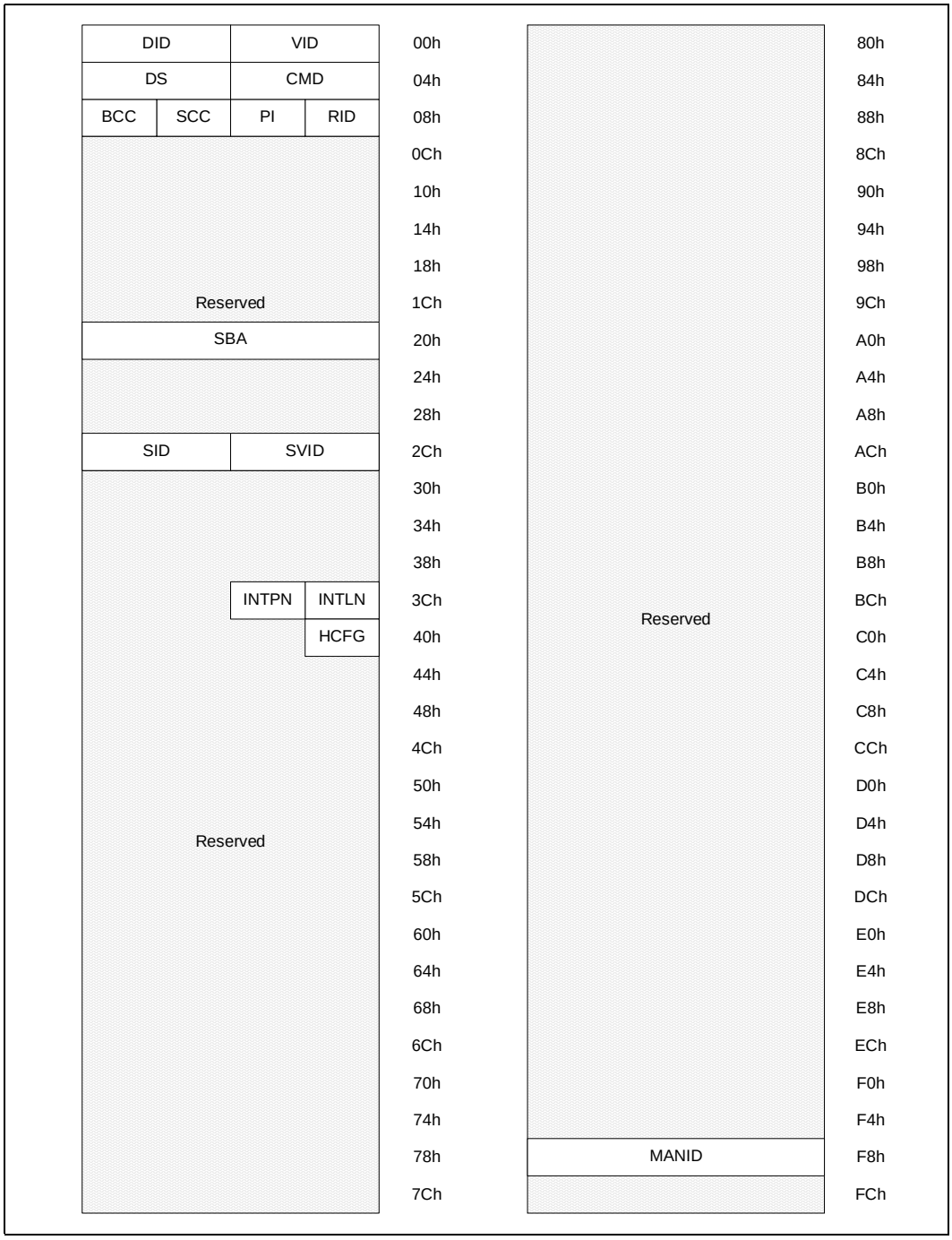
2.3.1.6 USB 1.1, Devices 0,1,2,3

Figure 12. USB I/O Registers

USBSTS	USBCMD		00h	Reserved	80h
FRNUM	USBINTR		04h		84h
FRBASEADD			08h		88h
Reserved		SOFMOD	0Ch		8Ch
PORTSC[0,1]			10h		90h
Reserved			14h		94h
			18h		98h
			1Ch		9Ch
			20h		A0h
			24h		A4h
			28h		A8h
			2Ch		ACh
			30h		B0h
			34h		B4h
			38h		B8h
			3Ch		BCh
			40h		C0h
			44h		C4h
			48h		C8h
			4Ch		CCh
			50h		D0h
			54h		D4h
			58h		D8h
			5Ch		DCh
			60h		E0h
			64h		E4h
			68h		E8h
			6Ch		ECh
			70h		F0h
			74h		F4h
			78h		F8h
			7Ch		FCh

2.3.1.7 SMBUS Device 31, Function 3

Figure 13. SMBUS Device 31, Function 3 Registers



2.3.1.8 SMB I/O

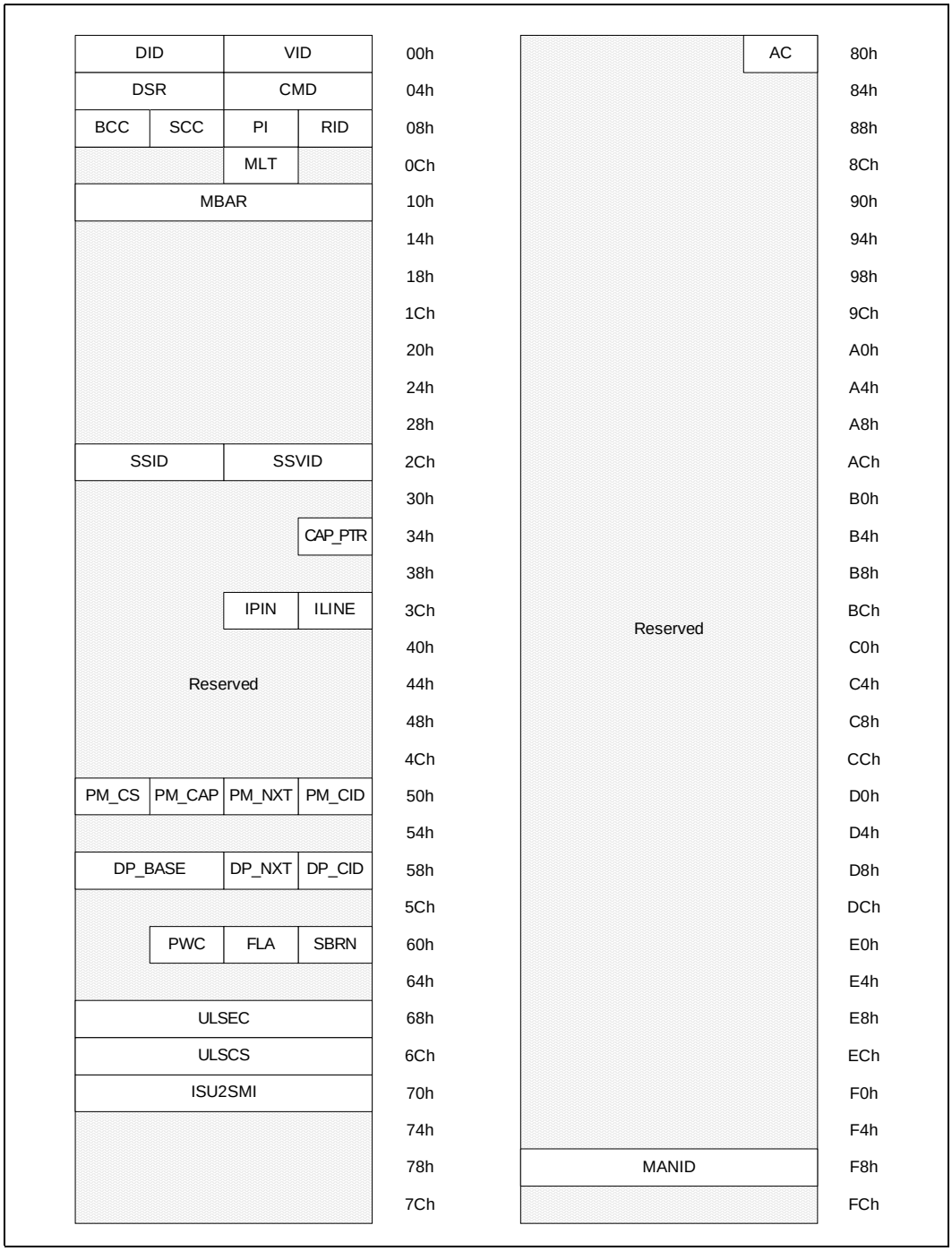
Figure 14. SMB I/O Registers

HCMD	HCTL	Reserved	HSTS	00h
HBD	HD1	HD0	TSA	04h
SD		RSA	PEC	08h
SMBC	SLPC	AUXC	AUXS	0Ch
		SCMD	SSTS	10h
NDHB	NDLB		NDA	14h
Reserved				18h
				1Ch
				20h
				24h
				28h
				2Ch
				30h
				34h
				38h
				3Ch
				40h
				44h
				48h
				4Ch
				50h
				54h
				58h
				5Ch
				60h
				64h
				68h
				6Ch
				70h
				74h
				78h
				7Ch

Reserved	80h
	84h
	88h
	8Ch
	90h
	94h
	98h
	9Ch
	A0h
	A4h
	A8h
	ACh
	B0h
	B4h
	B8h
	BCh
	C0h
	C4h
	C8h
	CCh
	D0h
	D4h
	D8h
	DCh
	E0h
	E4h
	E8h
	ECh
	F0h
	F4h
	F8h
	FCh

2.3.1.9 USB 2.0

Figure 15. USB 2.0 Registers



2.3.1.10 PCI Express Header Registers

This map includes registers from the following summary tables:

- Table 46, “PCI Express Header Registers Summary Table” on page 106
- Table 41, “SATA Register Summary: Additional SFF-8038i Configuration Registers” on page 104
- Table 42, “SATA Register Summary: PCI Power Management Capabilities Registers” on page 105
- Table 43, “SATA Register Summary: Message Signaled Interrupt Registers” on page 105
- Table 44, “SATA Additional Configuration Registers Summary” on page 105

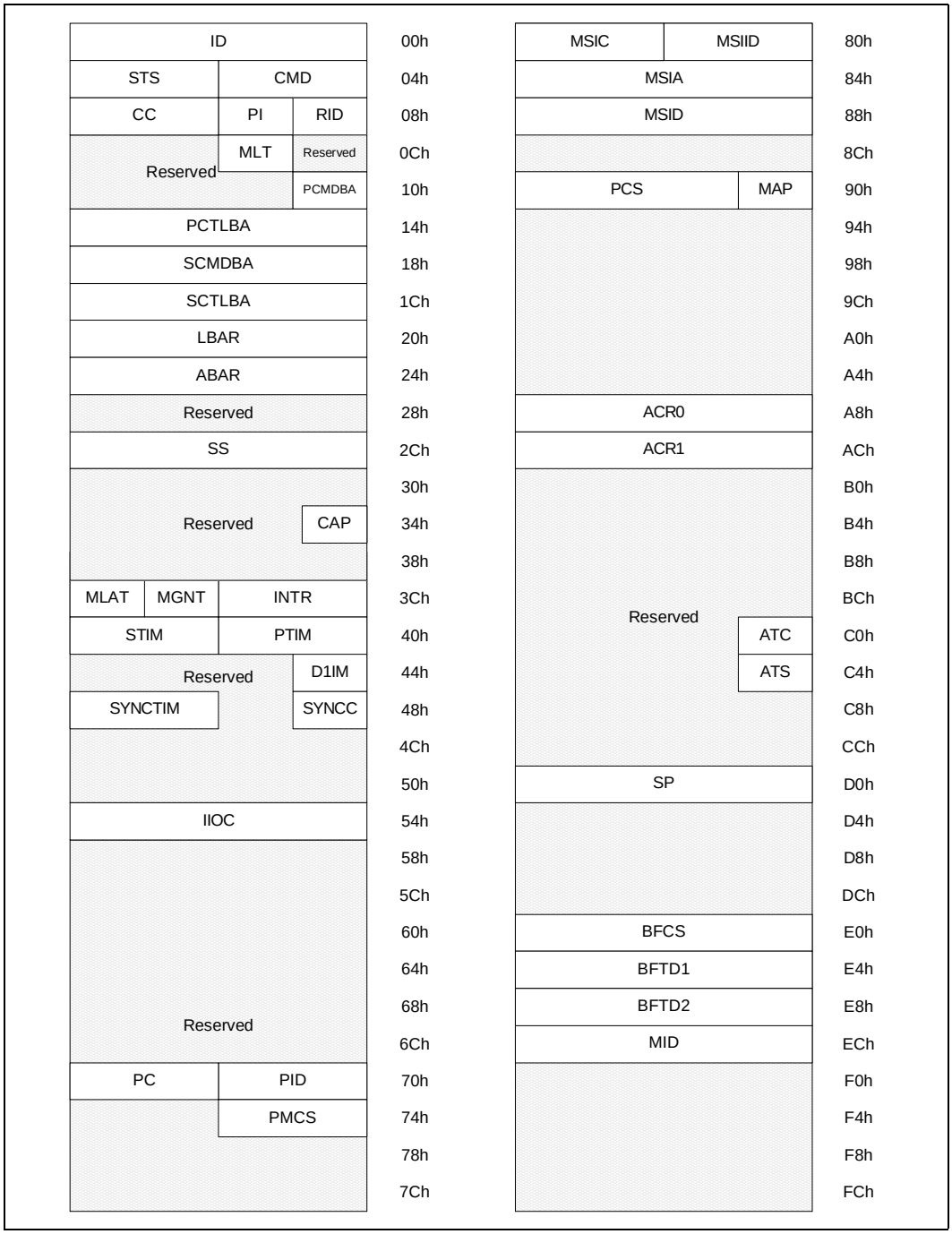
Figure 16. PCI Express Header Registers

ID				00h
STS		CMD		04h
CC			RID	08h
Reserved	HTYPE	MLT	CLS	0Ch
Reserved				10h
				14h
SLT	BNUM			18h
SSTS		IOBL		1Ch
MBL				20h
PMBL				24h
PMBU32				28h
PMLU32				2Ch
Reserved				30h
			CAPP	34h
				38h
BCTRL		INTR		3Ch
XCAP		CLIST		40h
DCAP				44h
DSTS		DCTL		48h
LCAP				4Ch
LSTS		LCTL		50h
SLCAP				54h
SLSTS		SLCTL		58h
		RCTL		5Ch
RSTS				60h
Reserved				64h
				68h
				6Ch
				70h
				74h
				78h
				7Ch

MC	MID	80h
MA		84h
	MD	88h
		8Ch
	SVCAP	90h
SVID		94h
		98h
		9Ch
PMC	PMCAP	A0h
PMCS		A4h
Reserved		A8h
		ACh
		B0h
		B4h
		B8h
		BCh
		C0h
		C4h
		C8h
		CCh
		D0h
		D4h
MPC		D8h
SMSCS		DCh
		E0h
		E4h
		E8h
		ECh
		F0h
		F4h
		F8h
		FCh

2.3.1.11 SATA

Figure 17. SATA Registers



2.3.2 IICH Memory Mapped Registers

Table 16. Bridging and Configuration Register (Memory Space) Summary Table (Sheet 1 of 2)

Address		Symbol	Register Name/Function	Default	Access
Start	End				
VC Configuration Registers					
0000h	0003Fh	VCH	Virtual Channel Capability Header Register	*10010002h*	RO
0004h	0007h	VCAP1	Virtual Channel Capability 1 Register	*0801h*	RO
0008h	000Bh	VCAP2	Virtual Channel Capability 2 Register	*0001h*	RO
000Ch	000Dh	PVC	Port Virtual Channel Control Register	0	RO, RW
000Eh	000Fh	PVS	Port Virtual Channel Status Register	0	RO
0010h	0013h	V0CAP	Virtual Channel 0 Resource Capability Register	0001h	RO
0014h	0017h	V0CTL	Virtual Channel 0 Resource Control Register	800000FFh	RO, RW
001Ah	001Bh	V0STS	Virtual Channel 0 Resource Status Register	0	RO
Root Complex Topology Configuration Registers					
0100h	0103h	RCTCL	Root Complex Topology Capability List Register	1A010005h	RO
0104h	0107h	ESD	Element Self Description Register	00000602h	RO, RWO
0110h	0113h	ULD	Upstream Link Descriptor Register	0001h	RO, RWO
0118h	011Fh	ULBA	Upstream Link Base Address Register	00000000_00000000h	RWO
0120h	0123h	RPB0D	Root PortB0 (PEB0) Descriptor Register	See desc	RO
0128h	012Fh	RPB0BA	Root PortB0 (PEB0) Base Address Register	00000000_00 0E0000h	RO
0130h	0133h	RPB1D	Root PortB1 (PEB1) Descriptor Register	See desc	RO
0138h	013Fh	RPB1BA	Root PortB1 (PEB1) Base Address Register	00000000_00 0E1000h	RO
0140h	0143h	RPB2D	Root Port B2 (PEB2) Descriptor Register	See desc	RO
0148h	014Fh	RPB2BA	Root Port B2 (PEB2) Base Address Register	00000000_00 0E2000h	RO
0150h	0153h	RPB3D	Root Port B3 (PEB3) Descriptor Register	See desc	RO
0158h	015Fh	RPB3BA	Root Port B3 (PEB3) Base Address Register	00000000_00 0E3000h	RO
Internal Link Configuration Registers					
01A0h	01A3h	ILCL	Internal Link Capability List Register	00010006h	RO
01A4h	01A7h	LCAP	Link Capabilities Register	0012441h	RO
01A8h	01A9h	LCTL	Link Control Register	0h	RO, RW
01AAh	01ABh	LSTS	Link Status Register	0041h	RO
I/O Data Bus Configuration Registers					
0224h	0227h	RPC	Root Port Configuration Register	00000000h	RO, RW
TCO Configuration Register					
3000h	3001Fh	TCTL	TCO Control Register	0h	RO, RW
Interrupt Configuration Registers					
3100h	3103h	D31IP	Device 31 Interrupt Pin Register	0042210h	RO, RW
3104h	3107h	D30IP	Device 30 Interrupt Pin Register	00002100h	RO
3108h	310Bh	D29IP	Device 29 Interrupt Pin Register	10004321h	RO, RW
310Ch	310Fh	D28IP	Device 28 Interrupt Pin Register	00004321h	RO, RW
3140h	3141h	D31IR	Device 31 Interrupt Route Register	03210h	RO, RW
3142h	3143h	D30IR	Device 30 Interrupt Route Register	03210h	RO, RW
3144h	3145h	D29IR	Device 29 Interrupt Route Register	03210h	RO, RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 16. Bridging and Configuration Register (Memory Space) Summary Table (Sheet 2 of 2)

Address		Symbol	Register Name/Function	Default	Access
Start	End				
3146h	3147h	D28IR	Device 28 Interrupt Route Register	03210h	RO, RW
31FFh	31FFh	OIC	Other Interrupt Control Register	0	RO, RW
General Configuration Registers					
3400h	3403h	RC	RTC Configuration Register	0	RO, RW, RWO
3404h	3407h	HPTC	High Performance Precision Timer Configuration Register	0	RO, RW
3410h	3413h	GCS	General Control and Status Register	See desc	RO, WO, RW
3414h	3417h	BUC	Backed-Up Control Register	See desc	RO, RW
3418hh	341Bh	FD	Function Disable Register	00000C12h	RO, RW
341C	341Fh	PRC	Power Reduction Control Register Clock Gating	0	RO, RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 17. High Precision Event Timers Registers Summary

Offset		Symbol	Register Name/Function	Default	Type
Start	End				
000h	007h	GCAP_ID	General Capabilities and ID Register	0429B17F 8086A201h	RO
010h	017h	GEN_CONF	General Configuration Register	00000000_ 00000000h	RW
020h	027h	GINTR_STA	General Interrupt Status Register	00000000_ 00000000h	RWC
0F0h	0F7h	MAIN_CNT	Main Counter Value Register	Xh	RW
100h	107h	TIM1_CONF	Timer 0 Configuration and Capabilities Register	Xh	RW
108h	10Fh	TIM1_COMP	Timer 0 Comparator Value Register	Xh	RW
120h	127h	TIM2_CONF	Timer 1 Configuration and Capabilities Register	Xh	RW
128h	12Fh	TIM2_COMP	Timer 1 Comparator Value Register	Xh	RW
140h	147h	TIM3_CONF	Timer 2 Configuration and Capabilities Register	Xh	RW
148h	14Fh	TIM3_COMP	Timer 2 Comparator Value Register	Xh	RW

Notes:
1. Reads to reserved registers or bits returns a value of 0.
2. Software must not attempt to lock the memory-mapped I/O ranges for High-Precision Timers. If attempted, the lock is not honored, which means potential deadlock conditions may occur.

Table 18. Patch Registers Summary Table (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
800	807	CTT0	Cycle Type Trigger Register 0	00h	RW and RO
808	80F	CTM0	Cycle Type Mask Register 0	00h	RW and RO
810	817	AT0	Address Trigger Register 0	00h	RW
818	81F	AM0	Address Mask Register 0	00h	RW
820	827	FDT0	First Data Trigger Register 0	00h	RW
828	82F	FDM0	First Data Mask Register 0	00h	RW
830	837	IST0	Internal State Trigger Register 0	00h	RW
838	83F	ISM0	Internal State Mask Register 0	00h	RW
840	87F		Trigger and Mask Registers 1 (see 000-03F for format)	00h	RW

Table 18. Patch Registers Summary Table (Sheet 2 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
880	8BF		Trigger and Mask Registers 2 (see 000-03F for format)	00h	RW
8C0	8FF		Trigger and Mask Registers 3 (see 000-03F for format)	00h	RW
900	BFF		Reserved for additional Trigger/Mask Registers (16 total)	00h	
C00	C03	PFCTRL0	Patch FIFO Control 0 Register	00h	RW
C04	C07	PFCTRL1	Patch FIFO Control 1 Register	00h	RW
C08	C3F		Patch FIFO Control [2:15] Register	00h	RW
C40	C7F		Reserved for more Patch FIFO Control Registers (32 total)	00h	
C80	C87	OPAR0	Operand Array Entry 0 Register	00h	RW
C88	C8F	OPAR1	Operand Array Entry 1 Register	00h	RW
C90	CBF		Operand Array Entries [2:7] Register	00h	RW
CC0	CFF		Reserved for more Operand Array Entries (16 total)	00h	
D00	D07	WSCT	Work Space Cycle Type Register	00h	RO
D08	D0F	WSA	Work Space Address Register	00h	RO
D10	D17	WSD	Work Space Data Register	00h	RO
D18	D1F		Reserved	00h	
D20	D27	BCT	Blocked Cycle Type Register	00h	RO
D28	D2F	BA	Blocked Address Register	00h	RO
D30	D37	BD	Blocked Data Register	00h	RO
D38	D3F	INTST	Internal State Register	00h	RO
D40	D47	GPCR	Global Patch Control Register	00h	RW
D48	D4F	SMIST	SMI Status Register	00h	RWC, RO
D50	DFE		Reserved		
DFF	DFF	CHKBIT	Future Chicken Bits Register	00h	RW
E00	E03	TRPST	Trap Status Register	00h	RWC
E04	E0F		Reserved	00h	
E10	E17	TRPC	Trapped Cycle Register	00h	RO
E18	E1F	TRPD	Trapped Write Data Register	00h	RO
E20	E7F		Reserved	00h	
E80	E87	TRPREG0	I/O Trap 0 Register	00h	RO, RW
E88	E8F	TRPREG1	I/O Trap 1 Register	00h	RO, RW
E90	E97	TRPREG2	I/O Trap 2 Register	00h	RO, RW
E98	E9F	TRPREG3	I/O Trap 3 Register	00h	RO, RW
EA0	FFF		Reserved		

Table 19. PCI to PCI Bridge Register Summary Table (Sheet 1 of 3)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
PCI Header					
00h	03h	ID	Identifiers Register	*244E8086h*	RO
04h	05h	CMD	PCI Command Register	0000h	RW, RO
06h	07h	PSTS	Primary Status Register	0010h	RWC, RO
08h	08h	RID	Revision Identification Register	See Description	RO
09h	0Bh	CC	Class Code Register	060401h	RO

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 19. PCI to PCI Bridge Register Summary Table (Sheet 2 of 3)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
0Dh	0Dh	PMLT	Primary Latency Timer Register	00h	RO
0Eh	0Eh	HEADTYP	Header Type Register	*01h*	RO
18h	1Ah	BNUM	Bus Number Register	000000h	RW, RO
1Bh	1Bh	SMLT	Secondary Master Latency Timer Register	00h	RW, RO
1Ch	1Dh	IOBASE_LIMIT	I/O Base and Limit Register	0000h	RW, RO
1Eh	1Fh	SSTS	Secondary Status Register	0280h	RWC, RO
20h	23h	MEMBASE_LIMIT	Memory Base and Limit Register	00000000h	RW, RO
24h	27h	PREF_MEM_ BASE_LIMIT	Prefetchable Memory Base and Limit Register	00010001h	RW, RO
28h	2Bh	PMBU32	Prefetchable Memory Upper 32 Bits Register	00000000h	RW
2Ch	2Fh	PMLU32	Prefetchable Memory Limit Upper 32 Bits Register	00000000h	RW
34h	34h	CAPP	Capability List Pointer Register	50h	RO
3Ch	3Dh	INTR	Interrupt Information Register	0000h	RW, RO
3Eh	3Fh	BCTRL	Bridge Control Register	0000h	RO, RW, RWC

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 19. PCI to PCI Bridge Register Summary Table (Sheet 3 of 3)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
Bridge Proprietary Configuration					
40h	41h	SPDH	Secondary PCI Device Hiding Register	*00h*	RW, RO
44h	47h	DTC	Delayed Transaction Control Register	00000000h	RW, RO
48h	4Bh	BPS	Bridge Proprietary Status Register	00000000h	RO, RWC
4Ch	4Fh	BPC	Bridge Policy Configuration Register	00000000h	RW, RO
PCI Bridge Vendor Capability					
50h	50h	SVCAP	Subsystem Vendor Capability Register	*000Dh*	RO
54h	54h	SVID	Subsystem Vendor IDs Register	00000000h	RW
Manufacturer's ID					
F8h	F8h	MANID	Manufacturer's ID Register	00010F80h	RO

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 20. Bridge Proprietary Configuration Registers Summary

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
40	41h	SPDH	Secondary PCI Device Hide Register	*00h*	RO, RW
44	47h	DTC	Delayed Transaction Control Register	*00000000h*	RO, RW
48	4Bh	BPS	Bridge Proprietary Status Register	*00000000h*	RO, RWC
4C	4Fh	BPC	Bridge Policy Configuration Register	*00000000h*	RO, RW

Table 21. LPC I/F – D31, F0 Configuration Registers Summary Table (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
PCI Configuration Registers					
00h	03h	ID	Vendor Identification Register	*2670h* *8086h*	RO
04h	05h	CMD	Device Command Register	*0007h*	RO, RW
06h	07h	STS	Status Register	*0200h*	RWC, RO
08h	08h	RID	Revision ID Register	01h	RW Once
09h	0Bh	CC	Class Code Register	060100h	RO
0Dh	0Dh	MLT	Master Latency Timer Register	00h	RO
0Eh	0Eh	HTYPE	Header Type Register	80h	RO
2Ch	2Fh	SID	Subsystem Identifiers Register	0000_0000	RW Only
ACPI/GPIO Configuration Registers					
40h	40h	ABASE	ACPI Base Address Register	00000001h	RO, RW
44h	44h	ACTL	ACPI Control Register	00h	RO, RW
48h	48h	GBA	GPIO Base Address Register	*00000001h*	RO, RW
4Ch	4Ch	GC	GPIO Control Register	00h	RO, RW
Interrupt Configuration Registers					
60h	60h	PARC	PIRQA Routing Control Register	80h	RW
61h	61h	PBRC	PIRQB Routing Control Register	80h	RW
62h	62h	PCRC	PIRQC Routing Control Register	80h	RW
63h	63h	PDRC	PIRQD Routing Control Register	80h	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 21. LPC I/F – D31, F0 Configuration Registers Summary Table (Sheet 2 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
64h	64h	SCNT	Serial IRQ Control Register	10h	RO, RW
68h	68h	PERC	PIRQE Routing Control Register	80h	RW
69h	69h	PFRC	PIRQF Routing Control Register	80h	RW
6Ah	6Ah	PGRC	PIRQG Routing Control Register	80h	RW
6Bh	6Bh	PHRC	PIRQH Routing Control Register	80h	RW
LPC I/O Configuration Registers					
80h	81h	IOD	I/O Decode Ranges Register	0000h	RO, RW
82h	83h	IOE	I/O Enables Register	0000h	RO, RW
84h	85h	LG1	LPC Generic Decode Range 1 Register	0000h	RO, RW
88h	88h	LG2	LPC Generic Decode Range 2 Register	0000h	RO, RW
Power Management Configuration Registers: Refer to Chapter 22.0, "Power Management"					
FWH Configuration Registers					
D0h	D3h	FS1	FWH ID Select 1 Register	00112233h	RO, RW
D4h	D5h	FS2	FWH ID Select 2 Register	4567h	RO, RW
D8h	DBh	FDE	FWH Decode Enable Register	FFCFh	RO, RW
DCh	DCh	BC	BIOS Control Register	00h	RO, RW Only
Root Complex Register Block Configuration Register					
F0h	F0h	RCBA	Root Complex Base Address Register	00000000h	RO, RW
Manufacturing Information Register					
F8h	F8h	MANID	Manufacturer's ID Register	00010F80h	RO

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 22. DMA Registers Summary (Sheet 1 of 2)

Port	Alias	Symbol	Register Name/Function	Default	Access[1]
00h	10h		Channel 0 DMA Base and Current Address Register	XXXX	RW
01h	11h		Channel 0 DMA Base and Current Count Register	XXXXh	RW
02h	12h		Channel 1 DMA Base and Current Address Register	XXXX	RW
03h	13h		Channel 1 DMA Base and Current Count Register	XXXXh	RW
04h	14h		Channel 2 DMA Base and Current Address Register	XXXX	RW
05h	15h		Channel 2 DMA Base and Current Count Register	XXXXh	RW
06h	16h		Channel 3 DMA Base and Current Address Register	XXXX	RW
07h	17h		Channel 3 DMA Base and Current Count Register	XXXXh	RW
08h	18h		Channel 0-3 DMA Command Register	000X0X00h	WO
08h	18h		Channel 0-3 DMA Status Register	XXXXXXXh	RO
0Ah	1Ah		Channel 0-3 DMA Write Single Mask Register	000001XXh	WO
0Bh	1Bh		Channel 0-3 DMA Channel Mode Register	000000XXh	WO
0Ch	1Ch		Channel 0-3 DMA Clear Byte Pointer Register	XXXXXXXXh	WO
0Dh	1Dh		Channel 0-3 DMA Master Clear Register	XXXXXXXXh	WO
0Eh	1Eh		Channel 0-3 DMA Clear Mask Register	XXXXXXXXh	WO
0Fh	1Fh		Channel 0-3 DMA Write All Mask Register	00001111b	RW
80h	90h[1]		Reserved Page Register	Undefined	
81h	91h[1]		Channel 2 DMA Memory Low Page Register	XXXXXXXh	RW

Notes:

1. Some registers are normally read-only, but are writable in Alt-Access mode. Likewise, there are some registers that are normally write-only, but are readable in Alt-Access mode. The individual register descriptions may not indicate this. See Alt-Access mode for more details.
2. The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 22. DMA Registers Summary (Sheet 2 of 2)

Port	Alias	Symbol	Register Name/Function	Default	Access[1]
82h	82h		Channel 3 DMA Memory Low Page Register	XXXXXXXh	RW
83h	93h[1]		Channel 1 DMA Memory Low Page Register	XXXXXXXh	RW
87h	97h[1]		Channel 0 DMA Memory Low Page Register	XXXXXXXh	RW
89h	99h[1]		Channel 6 DMA Memory Low Page Register	XXXXXXXh	RW
8Ah	9Ah[1]		Channel 7 DMA Memory Low Page Register	XXXXXXXh	RW
8Bh	9Bh[1]		Channel 5 DMA Memory Low Page Register	XXXXXXXh	RW
C4h	C5h		Channel 5 DMA Base and Current Address Register	XXXX	RW
C6h	C7h		Channel 5 DMA Base and Current Count Register	XXXXh	RW
C8h	C9h		Channel 6 DMA Base and Current Address Register	XXXX	RW
CAh	CBh		Channel 6 DMA Base and Current Count Register	XXXXh	RW
CCh	CDh		Channel 7 DMA Base and Current Address Register	XXXX	RW
CEh	CFh		Channel 7 DMA Base and Current Count Register	XXXXh	RW
D0h	D1h		Channel 4-7 DMA Command Register	000X0X00h	WO
D0h	D1h		Channel 4-7 DMA Status Register	XXXXXXXh	RO
D4h	D5h		Channel 4-7 DMA Write Single Mask Register	000001XXb	WO
D6h	D7h		Channel 4-7 DMA Channel Mode Register	000000XXb	WO
D8h	D9h		Channel 4-7 DMA Clear Byte Pointer Register	XXXXXXXXh	WO
DAh	DBh		Channel 4-7 DMA Master Clear Register	XXXXXXXXh	WO
DCh	DDh		Channel 4-7 DMA Clear Mask Register	XXXXXXXXh	WO
DEh	DFh		Channel 4-7 DMA Write All Mask Register	00001111b	RW

Notes:

1. Some registers are normally read-only, but are writable in Alt-Access mode. Likewise, there are some registers that are normally write-only, but are readable in Alt-Access mode. The individual register descriptions may not indicate this. See Alt-Access mode for more details.
2. The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 23. 8254 Timer Register Summary Table

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
40h	40h		Counter 0 Interval Time Status Byte Format Register	0XXXXXXXb	RO
40h	40h		Counter 0 Counter Access Ports Register	XXh	RW
41h	41h		Counter 1 Interval Time Status Byte Format Register	0XXXXXXXb	RO
41h	41h		Counter 1 Counter Access Ports Register	XXh	RW
42h	42h		Counter 2 Interval Time Status Byte Format Register	0XXXXXXXb	RO
42h	42h		Counter 2 Counter Access Ports Register	XXh	RW
43h	43h	TCW	Timer Control Word Register	XXXXXXXh	WO, RWS

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 24. APIC Indirect Registers (LPC I/F – D31, F0) Summary

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00	00	ID	Identification Register	0000h	RW
01	01	VS	Version Register	00170020	RO
02	0F	—	Reserved		

Table 24. APIC Indirect Registers (LPC I/F – D31, F0) Summary

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
10	11	REDIR_TBL0	Redirection Table 0 Register	XXXX00000_001XXXX	RW, RO
12	13	REDIR_TBL1	Redirection Table 1 Register		RW, RO
...	...	—	—		—
3E	3F	REDIR_TBL23	Redirection Table 23 Register		RW, RO
40F	FF	—	Reserved		

Table 25. APIC Direct Registers (LPC I/F – D31, F0) Summary

Address	Symbol	Register Name/Function	Default	Access
FEC0_0000h	IDX	Index Register	00h	RW
FEC0_0010h	DAT	Data	00h	RW
FEC0_0040h	EOI	EOI Register	00h	WO

Table 26. Processor I/F Registers Summary Table

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
61h	61h	NMI_SC	NMI Status and Control Register	00h	RW, RO
70h	70h	NMI_EN	NMI Enable (and Real Time Clock Index) Register	80h	RW (special)
92h	92h	PORT92	Fast A20 and Init Register	00h	RW
F0h	F0h	COPROC_ERR	Coprocessor Error Register	00h	WO
CF9	CF9h	RST_CNT	Reset Control Register	00h	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 27. Indexed Registers Summary

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
0Ah	0Ah	RTC_REGA	This register is used for general configuration of the RTC functions.	XXX	RW
0Bh	0Bh	RTC_REGB	This register is used for general configuration of the RTC functions.	X0X00XXX	RW
0Ch	0Ch	RTC_REGC	This register is used for general configuration of the RTC functions.	00X00000b (X: Undefined)	RO
0Dh	0Dh	RTC_REGD	This register is used for general configuration of the RTC functions.	10XXXXXXb (X: Undefined)	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 28. Summary Table for Power Management PCI Registers (PM — D31:F0)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
A0h	A1h	GEN_PMCON_1	General Power Management Configuration 1 Register (Core Well)	0200h	RW, RO, RWO
A2h	A2h	GEN_PMCON_2	General Power Management Configuration 2 Register (Resume Well)	00h	RW, RWC
A4h	A4h	GEN_PMCON_3	General Power Management Configuration 3 Register (RTC Well)	00h	RW, RWC

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 28. Summary Table for Power Management PCI Registers (PM — D31:F0)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
A9h	A9h	Cx-STATE_CNF	Cx State Configuration Register	00h	RW
AAh	AAh	C4-TIMING_CNT	C4 Timing Control Register	00h	RW
B8h	BBh	GPI_ROUT	GPI Route Control Register	00000000h	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 29. TCO I/O Registers Summary Table

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	01h	TRLD	TCO Timer Reload and Current Value Register	0000h	RW
02h	02h	TDI	TCO Data In Register (from the operating system to the SMI Handler)	00h	RW
03h	03h	TDO	TCO Data Out Register (from SMI Handler to the operating system)	00h	RW
04h	05h	TSTS1	TCO 1 Status Register	0000h	RW
06h	07h	TSTS2	TCO 2 Status Register	0000h	RW
08h	09h	TCTL1	TCO 1 Control Register	0000h	RW
0Ah	0Bh	TCTL2	TCO 2 Control Register	0008h	RW
0Ch	0Dh	TMSG	TCO Message 1 and 2 Register	00h	RW
0Eh	0Eh	TWDS	TCO Watchdog Status Register	00h	RW
0Fh	0Fh	RSV	Reserved	00h	
10h	10h	LE	Legacy Elimination Register	11h	RW
12h	13h	TTMR	TCO Timer Initial Value Register	0004h	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 30. GPIO Register Summary Table

GPIOBASE + Offset		Symbol	Register Name/Description	Default	Type
Start	End				
General Registers					
00h	03h	GPIO_USE_SEL	GPIO Use Select Register	1B0C01C0h	RW
04h	07h	GP_IO_SEL	GPIO Input/Output Select Register	E400 FFFFh	RW
0Ch	0Fh	GP_LVL	GPIO Level for Input or Output Register	FF3F0000h	RW
Output Control Registers					
18h	1Bh	GPO_BLINK	GPIO Blink Enable Register	0004 0000h	RW
Input Control Registers					
2Ch	2Fh	GPI_INV	GPIO Signal Invert Register	00000000h	RW
30h	33h	GPIO_USE_SEL2	GPIO Use Select 2 [63:32] Register	00000006h	RW
34h	37h	GP_IO_SEL2	GPIO Input/Output Select 2 [63:32] Register	00000300h	RW
38h	3Bh	GP_LVL2	GPIO Level for Input or Output 2 [63:32] Register	00030207h	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 31. USB I/O Registers Summary

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00	01h	USBCMD	USB Command Register	0000h	RW[1]
02	03h	USBSTS	USB Status Register	0020h	RWC
04	05h	USBINTR	USB Interrupt Enable Register	0000h	RW
06	07h	FRNUM	USB Frame Number Register	0000h	RW[1]
08	0Bh	FRBASEADD	USB Frame List Base Address Register	XXXXX000	RW
0C	0Ch	SOFMOD	USB Start of Frame Modify Register	40h	RW
10	13h	PORTSC[0,1]	Port [0,1] Status/Control Register	0080h	RWC[1]

Note:
1. These registers are WORD writable only. Byte writes to these registers have unpredictable effects.
2. The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 32. SMBUS Device 3, Function 3 Configuration Registers Summary

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00	01h	VID	Vendor ID Register	8086h	RO
02	03h	DID	Device ID Register	269Bh	RO
04	05h	CMD	Command Register	0000h	RW
06	07h	DS	Device Status Register	0280h	RW
08	08h	RID	Revision ID Register	See Desc.	RO
09	09h	PI	Programming Interface Register	00h	RO
0A	0Ah	SCC	Sub Class Code Register	05h	RO
0B	0Bh	BCC	Base Class Code Register	0Ch	RO
20	23h	SBA	Base Address Register	00000001h	RW
2C	2Dh	SVID	SVID Register	0000h	RO
2E	2Fh	SID	SID Register	0000h	RO
3C	3Ch	INTLN	Interrupt Line Register	00h	RW
3D	3Dh	INTPN	Interrupt Pin Register	See Desc.	RO
40	40h	HCFG	Host Configuration Register	00h	RW, RO
F8	FBh	MANID	Manufacturer's ID Register	00010F80h	RO

Notes:
1. Registers not listed in Table 32 are reserved. These will return 00h on reads and writes have no effect.
2. All the above registers are implemented in the core well.
3. The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 33. SMB I/O Registers Summary (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	00h	HSTS	Host Status Register	00h	RWC
02h	02h	HCTL	Host Control Register	00h	RW
03h	03h	HCMD	Host Command Register	00h	RW
04h	04h	TSA	Transmit Slave Address Register	00h	RW
05h	05h	HD0	Host Data 0 Register	00h	RW
06h	06h	HD1	Host Data 1 Register	00h	RW
07h	07h	HBD	Host Block Data Register	00h	RW
08h	08h	PEC	Packet Error Chec Register	00h	RW
09h	09h	RSA	Receive Slave Address Register	44h	RW

Table 33. SMB I/O Registers Summary (Sheet 2 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
0Ah	0Bh	SD	Slave Data Register	0000h	RO
0Ch	0Ch	AUXS	Auxiliary Status Register	00h	RW
0Dh	0Dh	AUXC	Auxiliary Control Register	00h	RW
0Eh	0Eh	SLPC	SM Link Pin Control <TCO Compatible mode only> Register	See Description	RW
0Fh	0Fh	SMBC	SM Bus Pin Control Register	See Description	RW
10h	10h	SSTS	Slave Status Register	00h	RWC
11h	11h	SCMD	Slave Command Register	00h	RW
14h	14h	NDA	Notify Device Address Register	00h	RO
16h	16h	NDLB	Notify Data Low Byte Register	00h	RO
17h	17h	NDHB	Notify Data High Byte Register	00h	RO

Table 34. SMB Slave Interface I/O Registers Summary

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
09h	09h	RSA	Receive Slave Address Register	44h	RW
0Ah	0Bh	SD	Slave Data Register	0000h	RO
10h	10h	SSTS	Slave Status Register	00h	RWC
11h	11h	SCMD	Slave Command Register	00h	RW
14h	14h	NDA	Notify Device Address Register	00h	RO
16h	16h	NDLB	Notify Data Low Byte Register	00h	RO
17h	17h	NDHB	Notify Data High Byte Register	00h	RO

Table 35. USB 2.0 Configuration Registers Summary Table (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Default	Special Notes	Access
Start	End					
USB 2.0 Configuration Registers						
00h	01h	VID	Vendor ID Register	8086h		RO
02h	03h	DID	Device ID Register	268Ch		RO
04h	05h	CMD	Command Register	0000h		RW
06h	07h	DSR	Device Status Register	0290h		RW
08h	08h	RID	Revision ID Register	See Desc		RO
09h	09h	PI	Programming Interface Register	20h		RO
0Ah	0Ah	SCC	Sub Class Code Register	03h		RO
0Bh	0Bh	BCC	Base Class Code Register	0Ch		RO
0Dh	0Dh	MLT	Master Latency Timer Register	00h		RO
10h	13h	MBAR	Memory Base Address Register	00000000h		RW
2Ch	2Dh	SSVID	Subsystem Vendor ID Register	XXXXh	no h/w reset	RWS
2Eh	2Fh	SSID	Subsystem ID Register	XXXXh	no h/w reset	RWS
34h	34h	CAP_PTR	Capabilities Pointer Register	50h		RO

Notes:

1. "Read/Write Special" means that the register is normally read-only but may be written when the WRT_RDONLY bit is set. Because these registers are expected to be programmed by BIOS during initialization, their contents must not get modified by HCRESET or D3-to-D0 internal reset.
2. All configuration registers in this section are in the core well unless otherwise noted. All configuration registers in this section are reset by the core well reset, and the D3-to-D0 warm reset, unless otherwise noted.
3. "Suspend" means that the register is implemented in the Suspend power well.
4. All bits or registers not listed are reserved. All reserved bits read as 0, but must be ignored by software.
5. All bits reset to 0 unless otherwise indicated.

Table 35. USB 2.0 Configuration Registers Summary Table (Sheet 2 of 2)

Offset		Symbol	Register Name/Function	Default	Special Notes	Access
Start	End					
3Ch	3Ch	ILINE	Interrupt Line Register	00h		RW
3Dh	3Dh	IPIN	Interrupt Pin Register	see register description		RO
50h	50h	PM_CID	Power Management Capability ID Register	01h		RO
51h	51h	PM_NEXT	Next Item Ptr #1 Register	58h		RWS
52h	53h	PM_CAP	Power Mgt Capabilities Register	C9C2h		RO, RWS
54h	55h	PM_CS	Power Mgt Control/Status Register	0000h	2 bits in Suspend	RW
58h	58h	DP_CID	Debug Port Capability ID Register	0Ah		RO
59h	59h	DP_NEXT	Next Item Ptr #2	00h		RO
5Ah	5Bh	DP_BASE	Debug Port Base Offset Register	20A0h		RO
60h	60h	SBRN	USB Release Number Register	20h		RO
61h	61h	FLA	Frame Length Adjustment Register	20h	Suspend	RW
62h	63h	PWC	Port Wake Capabilities Register	01FFh	Suspend	RW
68h	6Bh	ULSEC	USB 2.0 Legacy Support Extended Capability Register	00000001h	Suspend	RW
6Ch	6Fh	ULSCS	USB 2.0 Legacy Support Control/ Status Register	00000000h	Suspend	RW
70h	73h	ISU2SMI	Intel Specific USB 2.0 SMI Register	00000000h	Suspend	RW
80h	80h	AC	Access Control Register	00h		RW, RO
F8h	FBh	MANID	Manufacturer's ID Register	00010F80h		

Notes:

1. "Read/Write Special" means that the register is normally read-only but may be written when the WRT_RDONLY bit is set. Because these registers are expected to be programmed by BIOS during initialization, their contents must not get modified by HCRESET or D3-to-D0 internal reset.
2. All configuration registers in this section are in the core well unless otherwise noted. All configuration registers in this section are reset by the core well reset, and the D3-to-D0 warm reset, unless otherwise noted.
3. "Suspend" means that the register is implemented in the Suspend power well.
4. All bits or registers not listed are reserved. All reserved bits read as 0, but must be ignored by software.
5. All bits reset to 0 unless otherwise indicated.

Table 36. Host Controller Capability Registers Summary Table

MEM_BASE+ Offset		Symbol	Register Name/Function	Default	Special Notes	Access
Start	End					
00h	00h	CAPLENGTH	Capabilities Length Register	20h		RO
02h	03h	HCIVERSION	Host Controller Interface Version Number Register	0100h		RO
04h	07h	HCSPARAMS	Structural Parameters Register	00104208h	Suspend	RWS
08h	0Bh	HCCPARAMS	Capability Parameters Register	00006871h		RO

Notes:
1. "Suspend" means that the register is implemented in the Suspend power well

"Read/Write Special" means that the register is normally read-only, but may be written when the WRT_RDONLY bit is set. Because these registers are expected to be programmed by BIOS during initialization, their contents must not get modified by HCRESET or D3-to-D0 internal reset.

Table 37. Host Controller Operational Registers Summary Table

MEM_BASE + Offset		Symbol	Register Name/Function	Default	Special Notes	Access
Start	End					
20h	23h	USB 2.0CMD	USB 2.0 Command Register	00080000h		RW
24h	27h	USB 2.0STS	USB 2.0 Status Register	00001000h		RWC, RO
28h	2Bh	USB 2.0INTR	USB 2.0 Interrupt Enable Register	00000000h		RW
2Ch	2Fh	FRINDEX	USB 2.0 Frame Index Register	00000000h		RW, RO
30h	33h	CTRLDSSEG MENT	Control Data Structure Segment Register	00000000h		RW
34h	37h	PERIODICLI STBASE	Period Frame List Base Address Register	00000XXXh		RW
38h	3Bh	ASYNCLIST ADDR	Next Asynchronous List Address Register	00000000h		RW, RO
60h	63h	CONFIG FLAG	Configure Flag Register	00000000h	Suspend	RW, RO
64h	67h	PORTSC	Port 1 Status and Control Register	00003000h	Suspend	RW
68h	6Bh	PORTSC	Port 2 Status and Control Register	00003000h	Suspend	RW
6Ch	6Fh	PORTSC	Port 3 Status and Control Register	00003000h	Suspend	RW
70h	73h	PORTSC	Port 4 Status and Control Register	00003000h	Suspend	RW
A0h	B3h		Debug Port Registers (see Section 28.13.2, "Debug Port Registers")			RW

Table 38. Debug Port Registers Summary Table

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
A0h	A3h	CNTL_STS	Control/Status Register	00000000h	RW,RWC,RO,WO
A4h	A4h	USBPID	USB PIDs	0000000h	Bits 31:16 are Read-Only, Bits 15:00 are Read/Write
A8h	AFh	DATABUF	Data Buffer (Bytes 7:0)	00000000 00000000h	RW
B0h	B0h	CONFIG	Configuration Register	00007F01	RW

Table 39. I/O Registers Summary Table

Start	End	Symbol	Name	Default	Access
00h	00h	PCMD	Primary Command	00h	RO,RW
02h	02h	PSTS	Primary Status	00h	RO, RW, RWC
04h	07h	PDTP	Primary Descriptor Table Pointer	XXh	RO, RW
08h	08h	SCMD	Secondary Command	00h	RO,RW
0Ah	0Ah	SSTS	Secondary Status	00h	RO, RW, RWC
0Ch	0Fh	SDTP	Secondary Descriptor Table Pointer	XXh	RO, RW
10h	13h	INDEX	AHCI Index register	00000000h	RO, RW
14h	17h	DATA	AHCI Data register	See register description	RW

Table 40. SATA PCI Header Registers Summary Table

Offset		Symbol	Name	Default	Access
Start	End				
00h	03h	ID	Identifiers Register	See Desc 8086	RO
04h	05h	CMD	Command Register	0000h	RO, RW
06h	07h	STS	Device Status Register	02B0h	RO, RWC
08h	08h	RID	Revision ID Register	See Desc	RO
09h	09h	PI	Programming Interface Register	01h	RO
0Ah	0Bh	CC	Class Codes 00 Register	See register description	RO, RWOnce
0Dh	0Dh	MLT	Master Latency Timer Register	00h	RO
10h	10h	PCMDBA	Primary Command Block Base Address Register	00000001h	RO, RW
14h	17h	PCTLBA	Primary Control Block Base Address Register	00000001h	RO, RW
18h	1Bh	SCMDBA	Secondary Command Block Base Address Register	00000001h	RO, RW
1Ch	1Fh	SCTLBA	Secondary Control Block Base Address Register	00000001h	RO, RW
20h	23h	LBAR	Legacy Bus Master IDE Base Address Register	00000001h	RO, RW
24h	27h	ABAR	AHCI Base Address Register	00000000h	RO, RW
2Ch	2Fh	SS	Sub System Identifiers Register	00000000h	RO, RW
34h	34h	CAP	Capabilities Pointer Register	70h	RO
3Ch	3Dh	INTR	Interrupt Information Register	See register description	RO, RW

Table 41. SATA Register Summary: Additional SFF-8038i Configuration Registers

Offset		Symbol	Name	Default	Type
Start	End				
40h	41h	PTIM	Primary IDE Timing Register	*0000h*	RW
42h	43h	STIM	Secondary IDE Timing Register	0000h	RW
44h	44h	D1TIM	Slave IDE Timing Register	00h	RW
48h	48h	SYNCC	Synchronous DMA Control Register	00h	RO, RW
4Ah	4Bh	SYNCTIM	Synchronous DMA Timing Register	0000h	RO, RW
54h	57h	IIOC	IDE I/O Configuration Register	*00000000h*	RO, RW

Table 42. SATA Register Summary: PCI Power Management Capabilities Registers

Offset		Symbol	Name	Default	Access
Start	End				
70h	71h	PID	PCI Power Management Capability ID Register	*0001h*	RO
72h	73h	PC	PCI Power Management Capabilities Register	*4002h*	RO
74h	75h	PMCS	PCI Power Management Control and Status Register	*0000h*	RO, RW, RWC

Table 43. SATA Register Summary: Message Signaled Interrupt Registers

Offset		Symbol	Name	Default	Access
Start	End				
80h	81h	MSIID	Message Signaled Interrupt Identifiers Register	*7005h*	RO
82h	83h	MSIC	Message Signaled Interrupt Message Control Register	*0000h*	RO, RW
84h	87h	MSIA	Message Signaled Interrupt Message Address Register	*00000000h*	RO, RW
88h	89h	MSID	Message Signaled Interrupt Message Data Register	*0000h*	RW

Table 44. SATA Additional Configuration Registers Summary

Offset		Symbol	Name	Default	Type
Start	End				
90h	90h	MAP	Port Mapping Register	*00h*	RO, RW
91h	93h	PCS	Port Control and Status Register	*000000*	RO, RW, RWC
A0h	A0h	SIRI	SATA Indexed Register Index	00h	RO, RW
A4h	A7h	STRD	SATA Indexed Register Data	XXXXXXXXh	RW
A8h	ABh	ACR0	AHCI Capability Register 0	00100012h	RO
ACh	AFh	ACR1	AHCI Capability Register 1	00000048h	RO
C0h	C0h	ATC	APM Trapping Control Register	00h	RO, RW
C4h	C4h	ATS	APM Trapping Status Register	00h	RO, RWC
D0h	D3h	SP	Scratch Pad Register	00000000h	RW
E0h	E3h	BFCS	BIST FIS Control/Status Register	00000000h	RO, RW, RWC
E4h	E7h	BFTD1	BIST FIS Transmit Data, DW1 Register	00000000h	RW
E8h	EBh	BFTD2	BIST FIS Transmit Data, DW2 Register	00000000h	RW
F8h	F8h	MID	Manufacturer's ID Register	00010F80h	RO

Table 45. Generic Host Controller Register Summary Table

Start	End	Symbol	Description	Default	Access
00h	03h	CAP	Host Capabilities Register	C6127F05h	RO, RWOnce
04h	07h	GHC	Global Host Control Register	00000000h	RO, RW
08h	0Bh	IS	Interrupt Status Register	00000000h	RO, RWC
0Ch	0Fh	PI	Ports Implemented Register	00000000h	RO, RWOnce
10h	13h	VS	Version Register	00010100h	RO

Table 46. PCI Express Header Registers Summary Table

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	03h	ID	Identification Registers	See Desc8086h	RO
04h	05h	CMD	Command Register	0000h	RW, RO
06h	07h	STS	Device Status Register	00100h	RWC, RO
08h	08h	RID	Revision ID Register	See Desc	RO
09h	0Bh	CC	Class Code Register	060400h	RO
0Ch	0Ch	CLS	Cache Line Size Register	00h	RW
0Dh	0Dh	PLT	Primary Latency Timer Register	00h	RO
0Eh	0Eh	HTYPE	Header Type Register	81h	RO
18h	1Ah	BNUM	Bus Number Register	000000h	RW
1Bh	1Bh	SLT	Secondary Latency Timer Register	see desc	see desc
1Ch	1Dh	IOBL	I/O Base and Limit Register	0000h	RO, RW
1Eh	1Fh	SSTS	Secondary Status Register	0000h	RWC, RO
20h	23h	MBL	Memory Base and Limit Register	00000000h	RW, RO
24h	27h	PMBL	Prefetchable Memory Base and Limit Register	00010001h	RO, RW
28h	2Bh	PMBU32	Prefetchable Memory Base Upper 32-bits Register	00000000h	RW
2Ch	2Fh	PMLU32	Prefetchable Memory Limit Upper 32-bits Register	00000000h	RW
34h	34h	CAPP	Capability List Pointer Register	40h	RO
3Ch	3Dh	INTR	Interrupt Information Register	xx00h	RO, RW
3Eh	3Fh	BCTRL	Bridge Control Register	0000h	RO, RW

Table 47. PCI Express Register Summary: Root Port Capability Structure

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
40h	41h	CLIST	Capabilities List Register	8010h	RO
42h	43h	XCAP	PCI Express Capabilities Register	0041h	RO
44h	47h	DCAP	Device Capabilities Register	00000FE0h	RO
48h	49h	DCTL	Device Control Register	0000h	RO, RW
4Ah	4Bh	DSTS	Device Status Register	0010h	RO, RWC
4Ch	4Fh	LCAP	Link Capabilities Register	see desc	RO
50h	51h	LCTL	Link Control Register	0000h	RO, RW
52h	53h	LSTS	Link Status Register	see desc	RO
54h	57h	SLCAP	Slot Capabilities Register	00000060h	RO, RWO
58h	59h	SLCTL	Slot Control Register	0000h	RO, RW
5Ah	5Bh	SLSTS	Slot Status Register	0x000000h	RO, RWC
5Ch	5Dh	RCTL	Root Control Register	0000h	RO, RW
60h	63h	RSTS	Root Status Register	*00000000h*	RO

Table 48. PCI Express Register Summary: Message Signaled Interrupt Capability

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
80h	81h	MID	Message Signaled Interrupt Identifiers	*9005h*	RO
82h	83h	MC	Message Signaled Interrupt Control	*0000h*	RW, RO
84h	87h	MA	Message Signaled Interrupt Address	*00000000h*	RW, RO
88h	89h	MD	Message Signaled Interrupt Data	00h	RW, RO

Table 49. PCI Express Register Summary: PCI Bridge Vendor Capability

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
90h	91h	SVCAP	Subsystem Vendor Capability Pointer Register	*A00Dh*	RO
94h	97h	SVID	Subsystem Vendor IDs Register	*00000000h*	RWO

Table 50. PCI Express Register Summary: PCI Power Management Capability

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
A0h	A1h	PMCAP	Power Management Capability Pointer Register	*0001h*	RO
A2h	A3h	PMC	Power Management Capabilities Register	*C802h*	RO
A4h	A7h	PMCS	Power Management Control and Status Register	*00000000h*	RO

Table 51. PCI Express Register Summary: Port Configuration Capability

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
D8h	DBh	MPC	Miscellaneous Port Configuration Register	*00110000h*	RO, RW
DCh	DFh	SMSCS	SMI / SCI Status Register	*00000000h*	RO, RWC

Table 52. PCI Express Register Summary: VC Configuration Capability

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
100h	103h	VCH	Virtual Channel Capability Header Register	*18010002*	RO
104h	107h	VCAP1	Virtual Channel Capability 1 Register	00000001h	RO
108h	10Bh	VCAP2	Virtual Channel Capability 2 Register	*00000001h*	RO
10Ch	10Dh	PVC	Port VC Control Register	*0000h*	RO, RW
10Eh	10Fh	PVS	Port VC Status Register	*0000h*	RO
110h	113h	V0CAP	VC 0 Resource Capability Register	*00000001h*	RO
114h	117h	V0CTL	VC 0 Resource Control Register	*800000FFh*	RO, RW
11Ah	11Bh	V0STS	VC 0 Resource Status Register	*0000h*	RO

Table 53. PCI Express Register Summary: Advanced Error Reporting Capability

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
140h	143h	AECH	Advanced Error Reporting Capability Header Register	*0000_0000h*	RO
144h	147h	UES	Uncorrectable Error Status Register	*0000_0000h*	RO, RWC
148h	14Bh	UEM	Uncorrectable Error Mask Register	*0000_0000h*	RO, RWO
14Ch	14Dh	UEV	Uncorrectable Error Severity Register	*0006_0011h*	RO
150h	153h	CES	Correctable Error Status Register	0000_0000h	RO, RWC
154h	157h	CEM	Correctable Error Mask Register	0000_0000h	RO, RWC
158h	15Bh	AECC	Advanced Error Capabilities and Control Register	0000_0000h	RO
15Ch	16Bh	HL	Header Log Register	0000_0000h	RO
16Ch	16Fh	REC	Root Error Command Register	0000_0000h	RO
170h	173h	RES	Root Error Status Register	0000_0000h	RO

Table 54. PCI Express Register Summary: Root Complex Topology Capability Structure Registers

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
180h	183h	RCTCL	Root Complex Topology Capability List Register	*000010005h*	RO
184h	187h	ESD	Element Self Description Register	see bit desc	RO
190h	193h	ULD	Upstream Link Descriptor Register	*00000001h*	RO
198h	19Fh	ULBA	Upstream Link Base Address Register	see bit desc	RO

Table 55. I/O Register Summary Table

Offset	Register	Default	Type
Base+00h	Preload Value 1 Register 0	FFh	RW
Base+01h	Preload Value 1 Register 1	FFh	RW
Base+02h	Preload Value 1 Register 2	0Fh	RO, RW
Base+04h	Preload Value 2 Register 0	FFh	RW
Base+05h	Preload Value 2 Register 1	FFh	RW
Base+06h	Preload Value 2 Register 2	0Fh	RO, RW
Base+08h	General Interrupt Status Register	00h	RWC, RO
Base+0ch	Reload Register 0	00h	W
Base+0dh	Reload Register 1	00h	RO, RW
Base+10h	WDT Configuration Register 0	00h	RO, RW
Base+14h	Down Counter Register 0	00h	RO
Base+15h	Down Counter Register 1	00h	RO
Base+16h	Down Counter Register 2	00h	RO
Base+18h	WDT Lock Register	00h	RO, RW1, RW

Table 56. Configuration Register Summary (Sheet 1 of 2)

Global Configuration Registers			
Index	Type	Default	Configuration Register
07h	RW	00h	Logical Device Number
20h	R	00h	Device ID
21h	R	01h	Device Rev
28h	RW	01h	SIW I/F (wait states)
29h	RW	02h	SIRQ Configuration
2Eh	RW	00h	Test Mode Configuration Register
Logical Device 4 Registers (Serial Port 1)			
30h	RW	00h	Enable
60h	RW	00h	Base I/O Address MSB
61h	RW	00h	Base I/O Address LSB
70h	RW	00h	Primary Interrupt Select
74h	RW	04h	RSVD

Table 56. Configuration Register Summary (Sheet 2 of 2)

75h	RW	04h	RSVD
F0h	RW	00h	RSVD
Logical Device 5 Registers (Serial Port 2)			
30h	RW	00h	Enable
60h	RW	00h	Base I/O Address MSB
61h	RW	00h	Base I/O Address LSB
70h	RW	00h	Primary Interrupt Select
74h	RW	04h	RSVD
75h	RW	04h	RSVD
F0h	RW	00h	RSVD
Logical Device 6 Registers (Watchdog Timer)			
30h	RW	00h	Enable
60h	RW	00h	Base I/O Address MSB
61h	RW	00h	Base I/O Address LSB
70h	RW	00h	Primary Interrupt Select

3.0 Enhanced Direct Memory Access Controller (EDMA)

The IMCH includes an integrated four-channel Enhanced Direct Memory Access (EDMA) controller to facilitate “push model” block transfers without processor intervention for higher overall system performance. This section details the operating modes, setup, interfaces, register set, and high-level implementation of the EDMA controller. Details of the configuration registers associated with this feature are provided in “Device 1, Function 0: EDMA Registers” on page 394

3.1 Overview

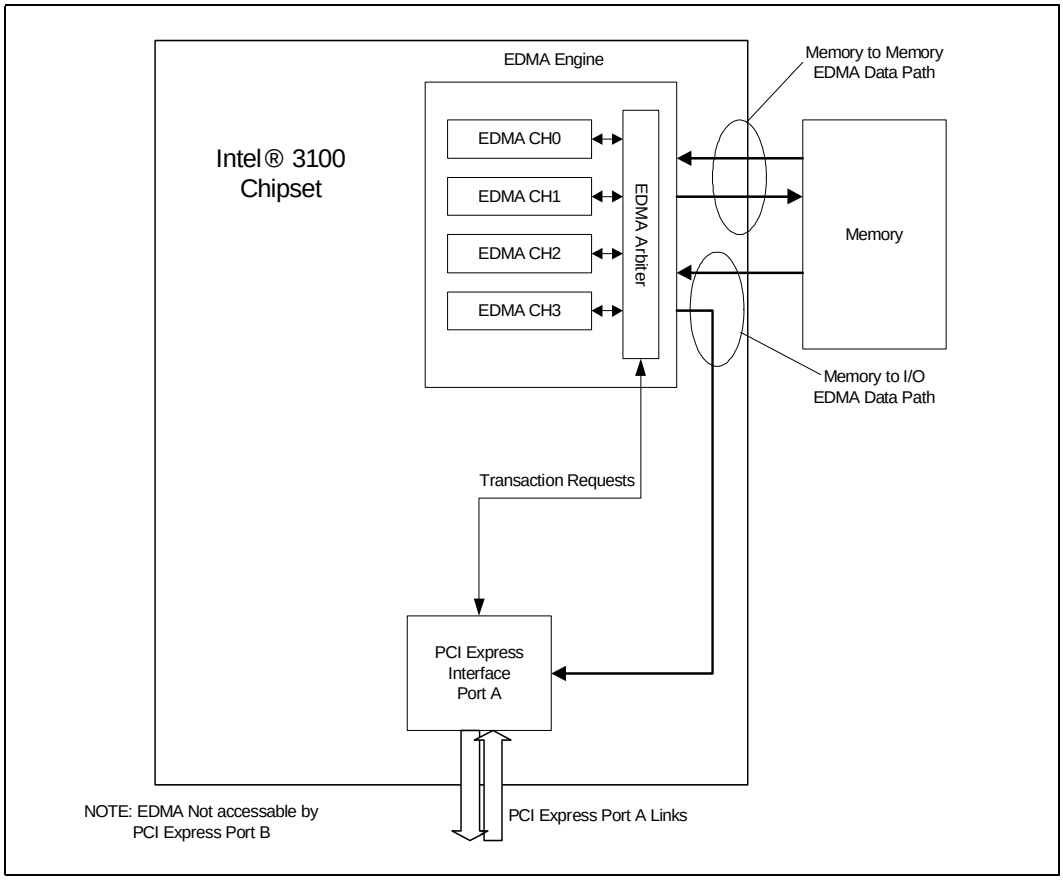
The EDMA engine provides a highly efficient means to move data within local system memory or from the local system memory to the I/O subsystem. Each EDMA channel provides low-latency, high-throughput data transfer capability with minimal processor intervention. For the processor, it is a “fire and forget” type of memory transfer, with a doorbell starting mechanism and interrupt capability for signaling completion.

Each channel optimizes block transfers of data through a linked-list descriptor chaining mechanism that supports scatter/gather operations. Each channel is responsible for providing the EDMA programming interface, executing the data transfers, and handling any errors encountered during operation.

Each channel initiates traffic on both the local system memory and outbound traffic arbiter interfaces, and is designed such that each independent channel is capable of generating at least 1 GB/s of traffic during data transfers. In the absence of competition from other traffic sources, multiple channels could theoretically saturate the local memory interface. See Figure 18.

Each channel is independently enabled by setting the Start bit in the Control Configuration Register. The Start bit is cleared after power-up or reset and consequently the EDMA controller is disabled until software explicitly turns each one on.

Figure 18. Concept Diagram of EDMA Data Path



3.1.1 Features

The following features are supported by the EDMA controller:

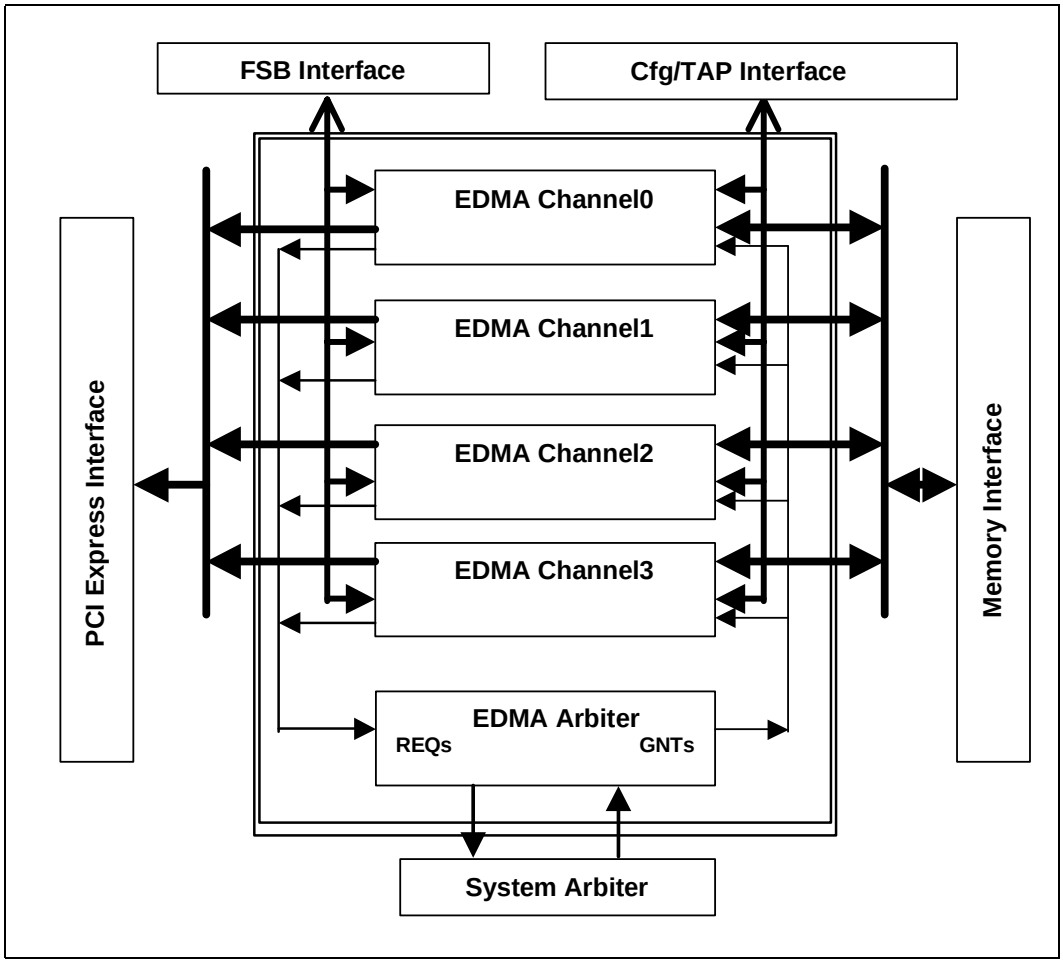
- Four independent channels (see Figure 19)
  - — Dedicated data transfer queue per channel
  - — Full register set for descriptor and transfer handling per channel
- Support for transfer between main memory locations and from memory to the I/O subsystem
- PCI Express* A port support of traffic class to provide external prioritization of traffic. The PCI Express B port is not supported as a target for the EDMA.
- Supports transfers only between two physical addresses
  - — 36-bit (64 GB) addressing range on the local system memory interface
  - — 36-bit addressing range on the Memory Mapped I/O Subsystem Interface (no NSI access)
- Maximum transfer of 16 Mbyte transfers per block
- Ability to transfer multiple blocks
- Fully programmable by the host processor
  - — Configuration space mapping for EDMA engine capability and control

  - Memory-mapped space for EDMA channel-specific register sets
- Chain Mode EDMA transfer with automatic data chaining for scattering/gathering of data blocks
  - EDMA chaining continued until a “null” descriptor pointer is encountered
  - Support for appending a block to the end of current EDMA chain
  - Automated descriptor retrieval from local memory during chaining – single read
- Programmable independent alignment between source and destination
  - Byte aligned transfer on the local system memory interface
  - Byte aligned transfer on the I/O subsystem interface
- Support for non-coherent transfers both to and from system memory on a per descriptor basis
  - Independent control of coherency for source and destination
- Programmable support for interrupt generation on block–by-block basis
  - Selectable MSI or legacy level-sensitive interrupt function
  - End of current block transfer
  - End of current chain
  - For any error causing a transfer to abort
- Increment of the source and destination address for standard transfers
- Increment of the destination and decrement of the source address to enable byte stream reversal
- Constant address mode for the destination address based on the transfer granularity to enable targeting of memory mapped I/O FIFO devices
- Buffer/memory initialization mode

3.1.2 Logical Block Diagram

Figure 19 shows the conceptual interface of the EDMA channels to different interfaces. The Cfg/TAP Interface is used to access internal registers.

Figure 19. Conceptual Diagram of Four Channel EDMA Engine



3.2 Channel Programming Interface

The EDMA channel programming interface is accessible from the processor via a combination of chain descriptors (shown in Figure 20) written to main memory and a memory-mapped internal register set. The EDMA controller provides four channels, each of which can be independently used to transfer data within the local system memory or from the local system memory to the I/O subsystem. Each channel has its own set of 12 registers. Refer to "Memory Mapped I/O for EDMA Registers" on page 511 for a description of the channel register set.

The channel programming interface is accessible from the processor via a combination of descriptors written to main memory and a memory-mapped internal register set. Each channel is programmed independently.

Each channel is programmed independently and supports full chaining capability. The chain descriptors can be cascaded together in system memory to form a linked list. Each chain descriptor contains all the information necessary for transferring a block of data, as well as a pointer to the next chain descriptor in the list. The next descriptor pointer of the last chain descriptor in a linked list will be a null pointer (address zero), indicating the end of that chain.

Throughout this document, all register references are made using the name of the lower 32-bit register irrespective of whether the target is a 32-bit or 64-bit register with lower and upper halves.

3.3 Chaining Operation

An EDMA access transfers a block of data from one address to another. The desired transfer is specified by setting up a linked list of chain descriptors in the local system memory, and initiated by programming the first chain descriptor start address into the Next Descriptor Address Registers (NDAR/NDUAR) of the EDMA channel and setting the Start bit of the Channel Control Register (CCR). Each block of the transfer is defined by a descriptor in main memory containing the source address, destination address, transfer length, and control values. Setting the Start bit of the Channel Control Register (CCR) causes the channel to fetch the current chain descriptor information and place it into its corresponding register set. Once all the register information has been fetched, the actual data transfer starts.

Note: The Start bit will be ignored unless the Channel Status Register (CSR) is in an appropriate state. Software must ensure that the status bits for end of chain, stopped, aborted, and active are all clear prior to attempting to initiate a new transfer with the start function.

3.3.1 Chain Descriptor Definition

All EDMA transfers are controlled by chain descriptors in the local system memory. A single block transfer will specify only a single chain descriptor. Chain descriptors can be linked together to form a linked list, providing a capability for complex EDMA scatter/gather operations.

Figure 20 shows the format of a chain descriptor. Each chain descriptor consists of eight contiguous DWords (32-bits) in the local system memory and must be naturally aligned to an eight Dword boundary. All eight DWords must be defined and are required for the proper operation of the EDMA engine.

Figure 20. Chain Descriptor in Memory

Chain Descriptor in Memory	*Description*
Source Address (SAR)	Lower 32-bit Source Memory Address Register
Source Upper Address (SUAR)	Upper 32-bit Source Memory Address Register
Destination Address (DAR)	Lower 32-bit Destination Memory Address Register
Destination Upper Address (DUAR)	Upper 32-bit Address of Next Chain Descriptor Register
Next Descriptor Address (NDAR)	Lower 32-bit Address of Next Chain Descriptor Register
Next Descriptor Upper Address (NDUAR)	Upper 32-bit Address of Next Chain Descriptor Register
Transfer Count (TCR)	Number of Bytes to Transfer Register
Descriptor Control (DCR)	Descriptor Control Register

B4483-01

3.3.2 EDMA Chain Descriptor in Memory

Each chain descriptor is composed of eight Dwords. Each Dword in the chain descriptor in the local memory is analogous to the corresponding EDMA channel register value. The bit definitions in the chain descriptor in memory are identical to those of the corresponding channel register. Refer to the EDMA channel-specific register definitions in "Memory Mapped I/O for EDMA Registers" on page 511 for descriptions of the fields defined by a chain descriptor.

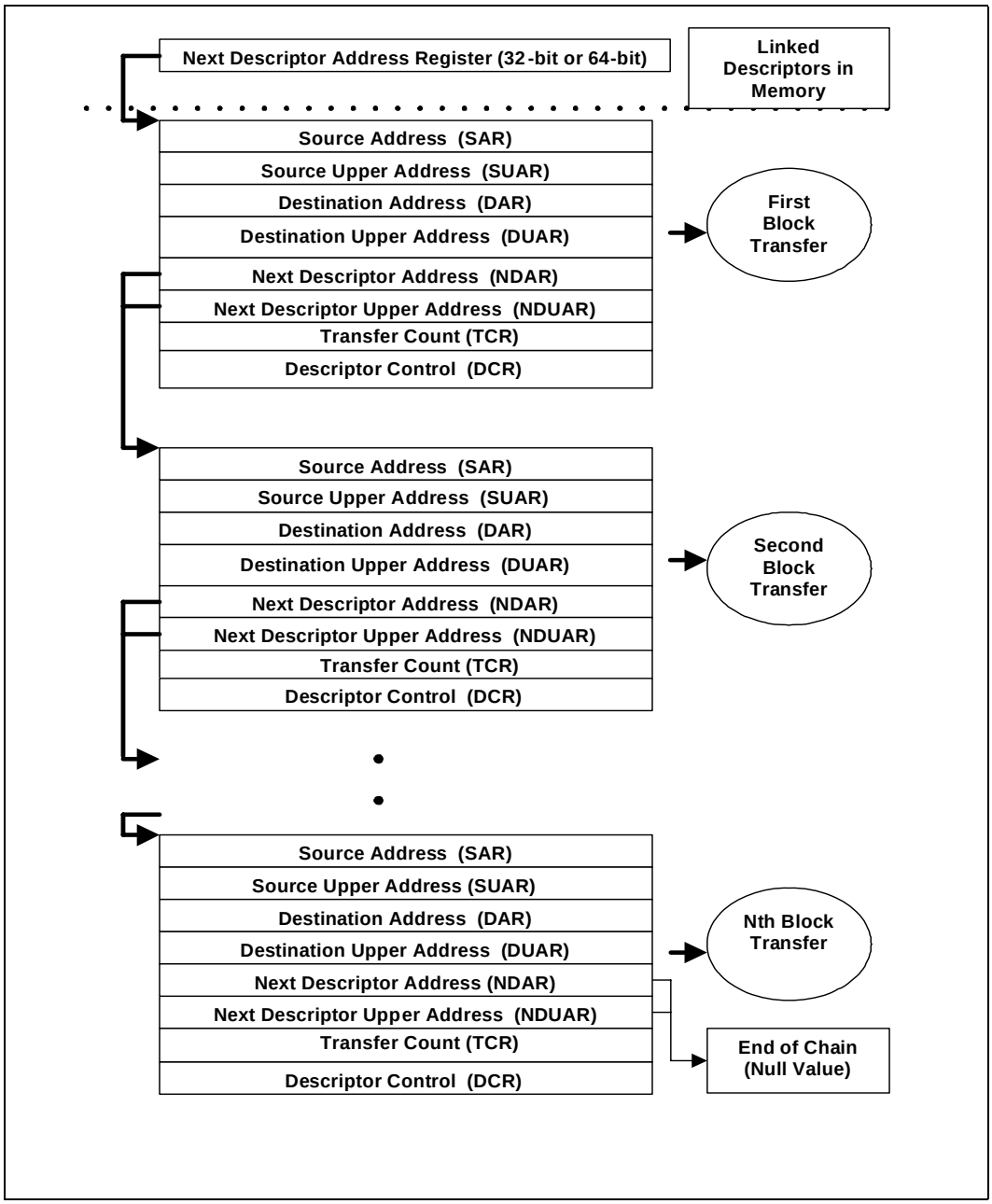
After a transfer has been requested, the EDMA channel reads the specified chain descriptor from local system memory and updates its own corresponding channel registers automatically.

3.3.3 Chain Descriptor Usage

A linked list of chain descriptors may be built in the local system memory to transfer data within the local system memory or from local system memory to I/O subsystem memory. An application may build multiple chain descriptors to transfer many blocks with differing source addresses, destination addresses, data transfer counts, alignments, and coherence attributes. The application may connect these chain descriptors using the Next Descriptor Address in a sequence of chain descriptors, creating a linked list of EDMA transfers, all of which may complete without any processor intervention. Figure 21 shows a linked list of transfers built in local system memory and illustrates how they are "chained" together.

It is possible but unexpected that the source and destination address ranges defined by chain descriptors may overlap in physical memory. While there are scenarios where this may produce no errors in the resulting memory image, it is left to software to ensure that no failure results from such usage. There are no hardware checks built into the EDMA mechanism to ensure that source and destination physical address ranges do not overlap. Similarly, there are no hardware interlocks to ensure that independent channels are not programmed to modify the same address range simultaneously. If software were to create such a situation, the resultant memory image would be indeterminate, since there are no guarantees as to the relative access ordering among simultaneously active channels.

Figure 21. Chaining Mechanism



3.3.4 Scatter/Gather Transfer

The EDMA descriptors in memory may be defined such that they cause the channel to perform typical scatter/gather data transfers. To “gather” data, software may create a linked list of descriptors that will move non-contiguous source blocks of data into a contiguous set of destination blocks. To “scatter” data, software may create a linked list of descriptors that will move contiguous blocks of source data to non-contiguous destination blocks. It is even possible to program the EDMA transfer descriptors in such a way that some blocks of source data within a single chain are moved to new memory locations, while other blocks are moved out to the I/O subsystem.

There is no hardware restriction limiting the nature of source and destination address ranges, other than that the source and destination types (memory or I/O subsystem) must match the descriptor address mappings. The IMCH aborts the EDMA operation and reports a programming error if this requirement is not met.

3.3.5 Appending to a Descriptor Chain

After an EDMA channel has started processing a linked list of descriptors, the application software may need to append a chain descriptor to the end of the current chain without tearing down the transfer in progress. Such an operation requires a mechanism to guarantee that a descriptor is never in the process of modification by the processor *while being retrieved* by the EDMA channel. (This would result in a hybrid descriptor loaded into the corresponding EDMA channel register set, and spurious operation.) The suspend function of the EDMA controller is defined to facilitate this type of usage.

The preferred mechanism for appending to a linked list currently being processed is to suspend the current transfer, modify the terminal chain descriptor to update its Next Descriptor Address field(s), and then allow the transfer to resume. This is accomplished by first setting the Suspend bit in the Channel Control Register (CCR) followed by a read of the Channel Status Register (CSR) to verify that the suspend has taken effect.The Next Descriptor Address fields of the terminal chain descriptor in the current linked list are updated with the address of the first descriptor to be appended. After this update has occurred, software then clears the Suspend bit, sets the Channel Resume bit and allows execution to proceed.

Note: A single write to the CCR may update both bits simultaneously.

This append algorithm covers all the following cases:

- The EDMA channel has completed execution of the terminal descriptor in the original chain, and is idle. The EDMA channel examines the Channel Resume bit when the Channel Control Register (CCR) is written. If the bit is set, the EDMA channel will automatically clear the bit and reread the last chain descriptor (as indicated by CDAR/CDUAR), which updates NDAR with the appended chain descriptor address. A non-null value in NDAR will result in a fetch of the target chain descriptor, and resumed execution. (If the resulting NDAR/NDUAR pair remains null, the EDMA channel will remain idle.)
- The EDMA channel is executing a descriptor prior to the terminal chain descriptor in the linked list. Regardless of whether the channel completes execution of its current descriptor prior to the CCR write to clear Suspend and set Channel Resume, the channel will re-read the current chain descriptor in response to the Channel Resume bit. The next chain descriptor in the chain will be fetched from the address indicated by NDAR/NDUAR, and the channel will continue execution. The appended chain descriptor (or descriptors) will be executed after the channel reaches the end of the original chain.
- The channel is executing the terminal chain descriptor at the time of the suspend command. The channel will complete the final chain descriptor of the original linked

list, and examine the state of the Channel Resume bit when the Suspend bit is cleared. As in the prior case, the channel will reread the current chain descriptor to update NDAR/NDUAR, load the first appended chain descriptor, and resume execution. The only difference in this case is that the reread operation on the current chain description was required for proper execution (in the prior case it was wasted effort, but did not result in erroneous behavior). If the channel had completed execution of the terminal chain descriptor and set the "end of chain" status bit, this bit is automatically cleared when the channel resumes operation.

- If the current transfer had a non-fatal error, it follows one of the above cases. If the error is fatal, the channel will abort, and it is up to software to take proper action and restart the EDMA transfer. Note that the channel will ignore the state of the Channel Resume bit if the abort status has not been cleared from the CSR. This simplifies the case of linked list append, as software need not take extra steps to verify that no errors exist prior to setting the Channel Resume bit. The normal polling or interrupt mechanism may handle the error without interacting with the append routine.

Note: Software is at liberty to modify the Next Descriptor Address fields of the terminal chain descriptor at any time after setting the Suspend bit in the CCR – there is no requirement that software needs to verify that the channel has gone idle prior to modifying the memory image. Also, software does not need to verify that the channel has completed execution of the current chain descriptor and acknowledged Suspend EDMA prior to issuing the final update to CSR that sets the Channel Resume bit. The hardware interlock will cover the case where the end of the chain descriptor is reached during the append sequence, but proper operation is guaranteed regardless of whether the interlock is exercised.

A further simplification to the linked list append sequence is possible in the case of chain descriptors located strictly below the 4 GB boundary in memory; that is, in the case where NDUAR of the terminal descriptor is zero and only NDAR contains asserted bits. Under these conditions, it is safe to issue the NDAR write cycle without first suspending operation, because there is no risk of a hybrid NDAR/NDUAR pair retrieved by the channel. If desired, software could take the simplified approach of issuing the descriptor update followed by a CCR write to set the Channel Resume bit. In all cases, this will result in successful execution of the appended chain irrespective of current execution status.

3.3.6 Splicing a Descriptor Chain into a Linked List

Software may utilize a slight modification of the algorithm described in "Appending to a Descriptor Chain" on page 117 to splice a new descriptor or chain of descriptors into the chain already executing. Such an operation would be useful to provide service to a higher priority EDMA transfer without aborting work already in progress.

The set of steps required to splice into a chain are as follows:

1. Write to the CCR to set the Suspend EDMA bit.
2. Read the CDAR/CDUAR pair to determine which chain descriptor the EDMA channel is currently executing.
3. Read the Next Descriptor Address field of the current chain descriptor, and write the retrieved address into the Next Descriptor Address field of the terminal chain descriptor in the linked list to be spliced-in.
4. Write the address of the descriptor (or lead descriptor of the chain) to be spliced-in into the Next Descriptor Address field of the current descriptor (in memory).
5. Write to the CCR to clear the Suspend bit and set the Channel Resume bit.

The hardware interlock of the suspend function will guarantee that the channel will not proceed beyond the current chain descriptor until the Suspend bit has been cleared in the CCR, and the Channel Resume function will guarantee that the current chain descriptor will be reread to retrieve the modified NDAR/NDUAR value pointing to the spliced chain descriptor. The channel will resume execution with the head of the spliced linked list, and will traverse that linked list back to its original list of chain descriptors.

Note: The channel must refrain from updating CDAR/CDUAR from NDAR/NDUAR in addition to dropping the returned data. Were the channel to update its CDAR/CDUAR values, it would "skip" the entire spliced chain in response to Channel Resume, because software would have spliced in the new descriptor chain at a position "behind" the new value for CDAR/CDUAR in the original chain.

3.4 Transfer Types

The EDMA controller is optimized to perform high throughput data transfers between local memory locations, and from local memory to I/O subsystem memory. Supported transfer types are summarized in the following sections.

3.4.1 Local Memory to Local Memory

The local memory to local memory transfer will move blocks of data specified by descriptors from one region in main memory to another. The channel control hardware will issue read cycles to the local memory interface using an incrementing or decrementing source address, and place the retrieved data into a channel buffer. It will then issue write cycles back to the memory interface using an incrementing or constant destination address. Each EDMA channel supports pipelining making it possible for a single channel to have multiple read and write cycles active at the same time.

All read requests to memory are a full cache-line in length (64 B), so the EDMA channel must discard data as needed to realign the initial read in a transfer to the alignment specified by the source address. The number of cache-line reads issued by the controller in any given internal arbitration cycle is dependent upon the number of available cache-line spaces in the data queue, and upon the configuration of the inbound/outbound arbiter, but is limited to a maximum of two cache-line requests.

All write requests to memory are also a full cache-line in length, although not all bytes must be enabled for every write. The EDMA channel is responsible for translating the alignment specified by the destination address registers and the destination alignment bit in DCR into a corresponding set of byte enables for the initial write in a transfer. Once the initial alignment has been enforced, the rest of the transfer on behalf of any given descriptor is contiguous.

3.4.2 Local Memory to I/O Subsystem Memory

The local memory to I/O memory transfer will move blocks of data specified by chain descriptors from a source region in main memory to a destination region in the I/O subsystem. The channel control hardware will issue read cycles to the local memory interface using an incrementing or decrementing source address, and place the retrieved data into a channel buffer. It will then issue write cycles back to the memory interface using an incrementing or constant destination address. Each EDMA channel supports pipelining for this transfer type as well, thus multiple read and write requests may be outstanding from the same EDMA channel at any given time during a block transfer.

All read and write requests are full cache-line size (64 B), irrespective of alignment and length specified in the descriptors; it is up to the EDMA channel to discard read data and calculate write byte enables to enforce the descriptor alignment specified.

The number of reads issued by each EDMA channel in any given internal arbitration cycle is dependent upon the number of available cache-line spaces in the data queue, and upon the configuration of the inbound/outbound arbiter, but is limited to a maximum of two cache-line requests. The number of writes issued in any given arbitration cycle is dependent upon the number of cache-lines waiting in the data queue, and upon the CCR configuration, but is limited to a maximum of two cache-line requests. The EDMA controller never speculatively issues a write in anticipation of data returning from the memory subsystem.

3.4.3 I/O Memory to Local Memory

The I/O memory to local memory transfer is not supported.

3.4.4 I/O Memory to I/O Memory

The I/O memory to I/O memory transfer is not supported.

3.5 Addressing

Each EDMA channel is capable of 64-bit addressing on both source and destination interfaces, although it will internally truncate all addresses to 36 bits.

Note: If any bit above bit 35 is asserted, this truncation causes an abort; an error will be logged in the CSR.

Alignment specification is independent for source and destination. Transfers may be specified to be aligned to any byte boundary except in destination constant address modes where the granularity is greater than 1-byte.

Each EDMA channel uses direct addressing for both the source and destination interfaces. There is no internal support for any virtual address translation.

Each EDMA channel will attempt to compensate for misalignment between source and destination. At a minimum, misalignment will result in decreased performance at either end of the transfer; where a second read is required prior to the first write, or vice-versa.

3.5.1 Address Coherence

Each EDMA channel provides support for non-coherent access specification to improve bandwidth and provide more consistent average latency, as well as to free the FSB for simultaneous processor traffic. The source and destination addresses for each EDMA channel may be independently specified on a chain descriptor granularity via bit settings in the DCR to be either coherent or non-coherent. For non-coherent accesses, no FSB snoop cycle is issued on behalf of EDMA memory accesses to snoop processor caches. It is up to software to verify that snoops of processor caches are not required for proper system operation prior to setting either of the non-coherent bits in any given descriptor. Non-coherent accesses are used for un-cacheable memory regions, or for cacheable regions where software can guarantee no modified state in any processor cache by some other means.

The non-coherent attribute further implies relaxed posted write ordering as defined by PCI/PCI-X. A non-coherent write may pass coherent posted writes en-route to memory. Software should verify that snoops of processor caches are not required for proper system operation prior to setting either of the non-coherent bits in the DCR field of any given descriptor. Software need not take special steps to accommodate the relaxed ordering behavior, because each channel will only generate a single stream of output per descriptor, and no ordering is defined between competing I/O subsystem traffic

sources. Non-coherent access may be used for uncacheable memory regions, or for cacheable regions where software can guarantee the processor cache state has not been modified by other means.

An example usage model for non-coherent accesses is management of data block areas reserved for use by an EDMA-capable peripheral device, such as a network interface controllers (NIC). The NIC writes data directly into a buffer in memory allocated for the exclusive use of that device. Each EDMA channel would then execute a block transfer from that buffer to an area allocated for processor use. Both integrity verification and security functions executed by the processor could follow such a model. As long as software ensures that the processor never traverses the device-allocated memory, the block transfer could be accomplished using non-coherent source address reads followed by coherent destination address writes.

Note: I/O subsystem destination addresses are always treated as non-coherent or coherent based on the bit setting in the DCR. Setting the destination coherency bit will result in the PCI Express snoop not required attribute bit being clear, snoop required.

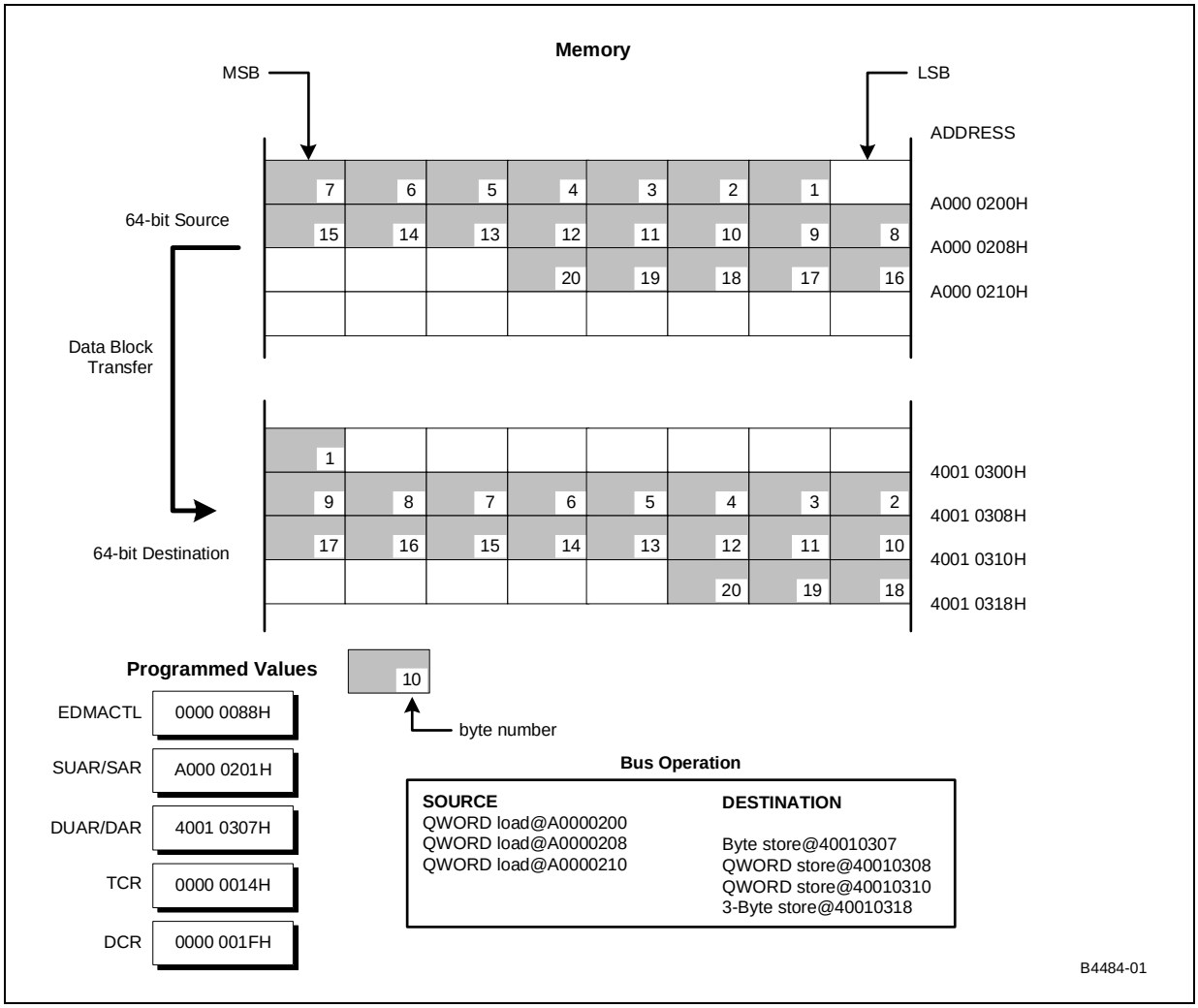
3.5.2 Addressing Modes

Many different addressing modes are available, including standard byte movement mode, byte reversal mode, constant address mode, and memory and buffer initialization modes. In the examples shown for each of the following modes, a 64-bit interface is used for simplicity. The interface could be the memory interface or an external device on an expansion bus. Internally, the EDMA data path is significantly wider.

3.5.2.1 Standard Byte Movement Mode

This mode is the most common method data is transferred within memory sub-system. In this mode, the source and destination are specified down to the byte address. The source address is incremented as data is read and the destination address is incremented as data is written. Transfers can be memory to memory or memory to memory mapped I/O. Figure 22 illustrates a memory to memory data transfer between unaligned 64-bit, source and destination addresses.

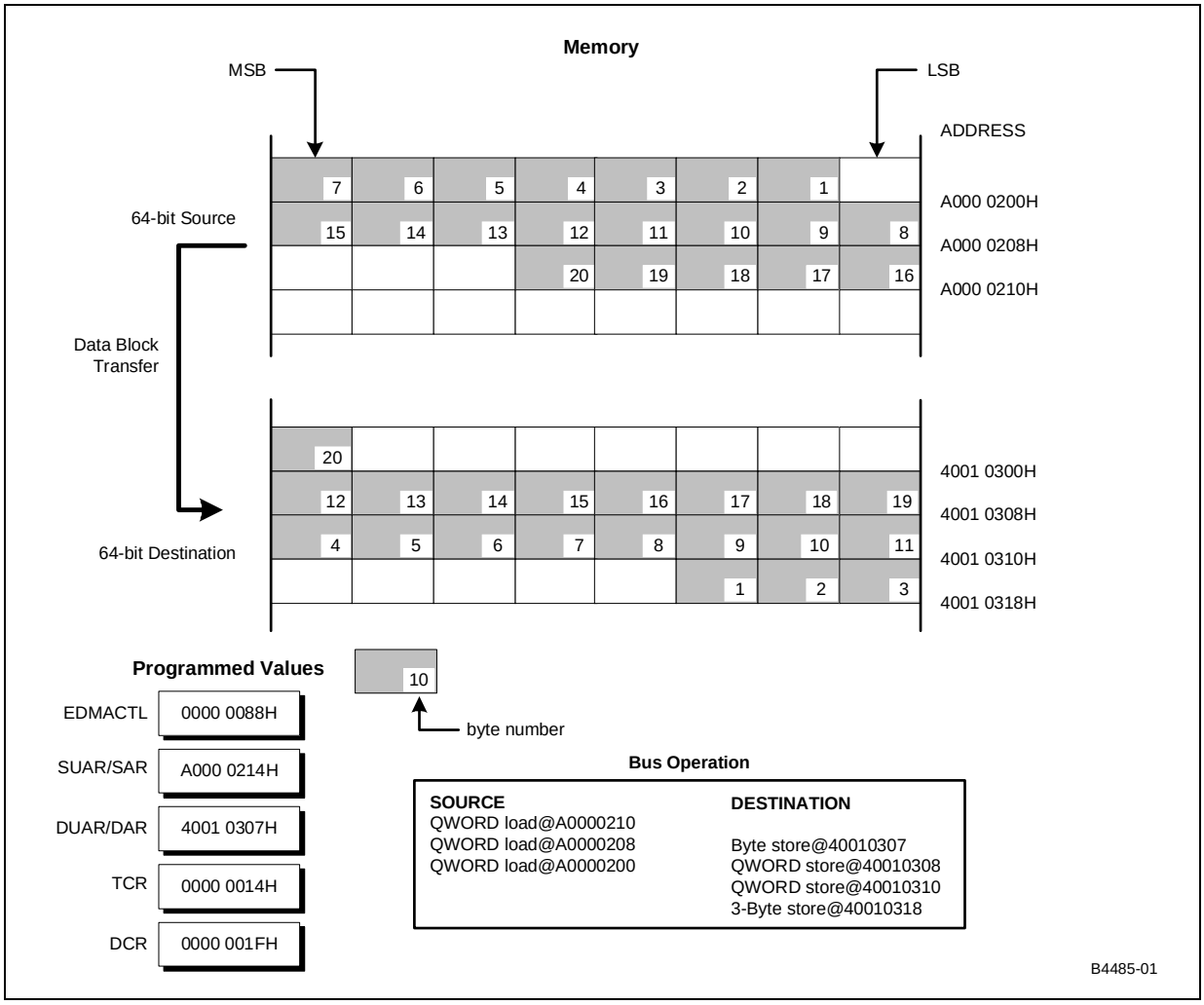
Figure 22. Source and Destination in Increment Mode Transfer



3.5.2.2 Decrement/Byte Reversal Mode

This mode is useful when an entire data stream needs to be reversed at the byte level. This must not be confused with endian swapping as this implies a specific word size. In this mode, the source and destination are specified down to the byte address. The source data is read in reverse order and written to the destination in increasing order. Transfers can be memory to memory or memory to memory mapped I/O. Figure 23 illustrates a memory to memory data transfer between unaligned 64-bit, source and destination addresses when the source is in decrement mode and the destination is in increment mode.

Figure 23. Source in Decrement and Destination in Increment Mode Transfer (Byte Reversal)



3.5.2.3 Constant Address Modes

In constant addressing mode, there is built-in support for “mailbox” destinations in the memory mapped I/O subsystem. A mailbox is a single or limited set of addresses used to collect information for dispersal later to their actual destination addresses by the receiving device. In Constant Address mode, one, two, or four bytes will be sent repeatedly until the byte count is satisfied.

The source address can be byte aligned; however, unlike other transfer modes, in Constant Address Mode, the destination address must be aligned to the granularity size. No errors will be flagged if the destination address is not matched to the granularity but the required lower address bits will be ignored. Additionally, software must ensure that the transfer byte count is an integer multiple of the granularity size. No error will be flagged if the transfer count is not an integer multiple and the remaining bytes in the requested granularity will be padded and transferred.

In this mode, the EDMA will write to the address that is placed in the DAR/DUAR pair with the transfer size specified in the granularity field. The transfers will continue until the byte count register is satisfied. Logically, transfers will only be memory to memory mapped I/O.

Constant Address mode can be used with an increasing or decreasing source address to present data to the external device in either order.

The Figure 24 though Figure 26 illustrate data transfers between an unaligned 64-bit, source and a constant destination addresses when the source is in increment mode.

Figure 24. Source in Increment and Destination in 1-Byte Granularity Constant Mode Transfer

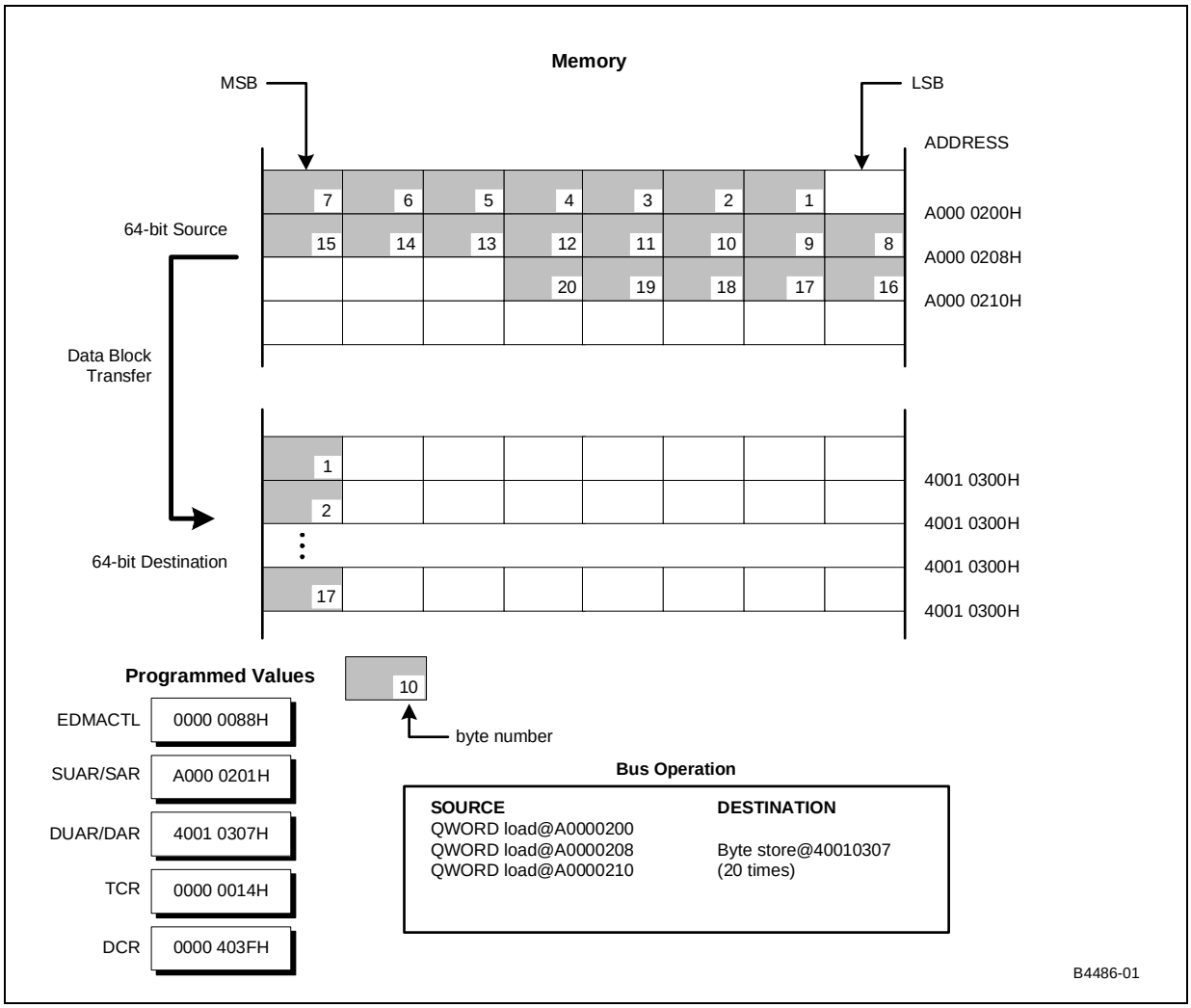


Figure 25. Source in Increment and Destination in 2-Byte Granularity Constant Mode Transfer

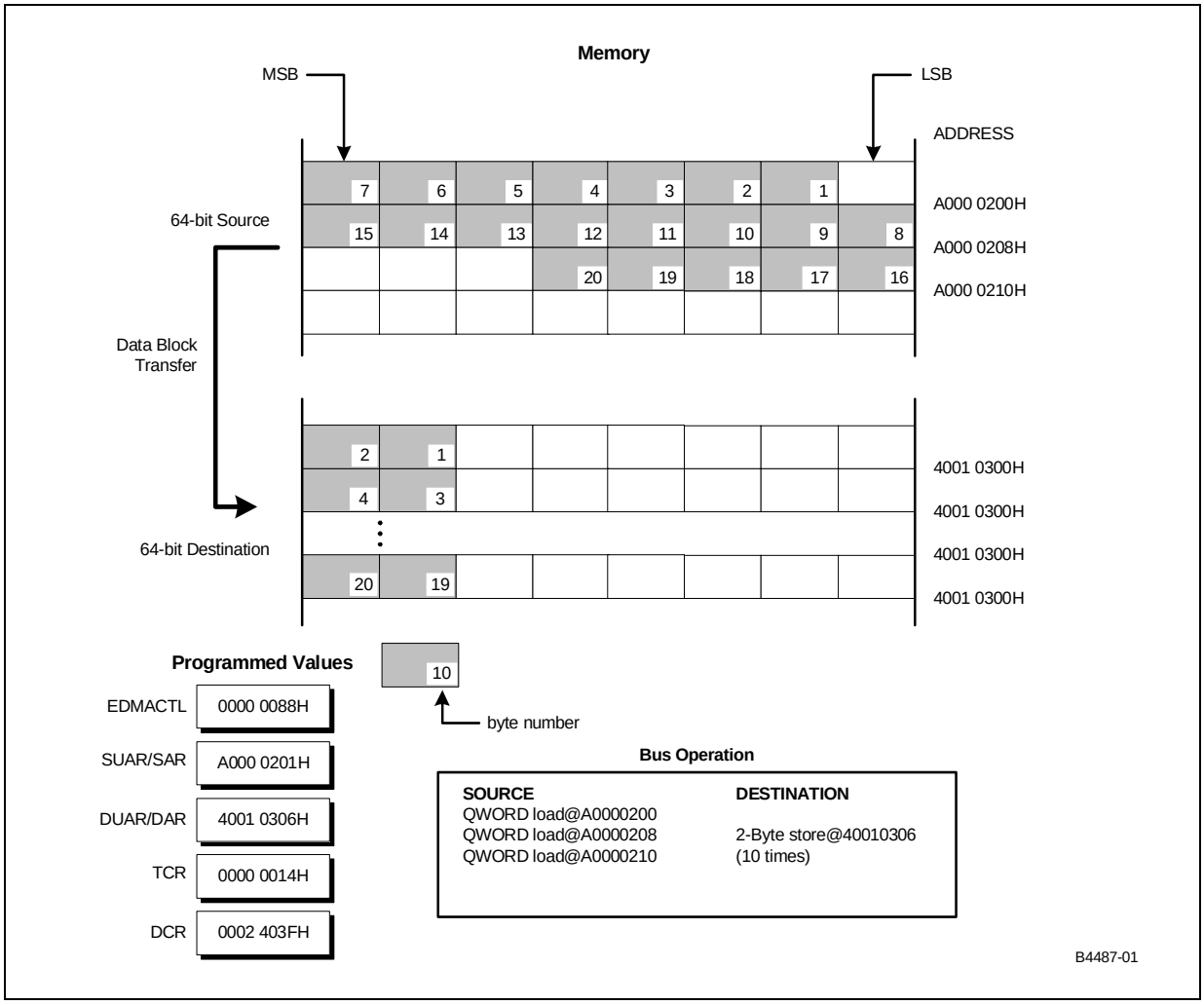
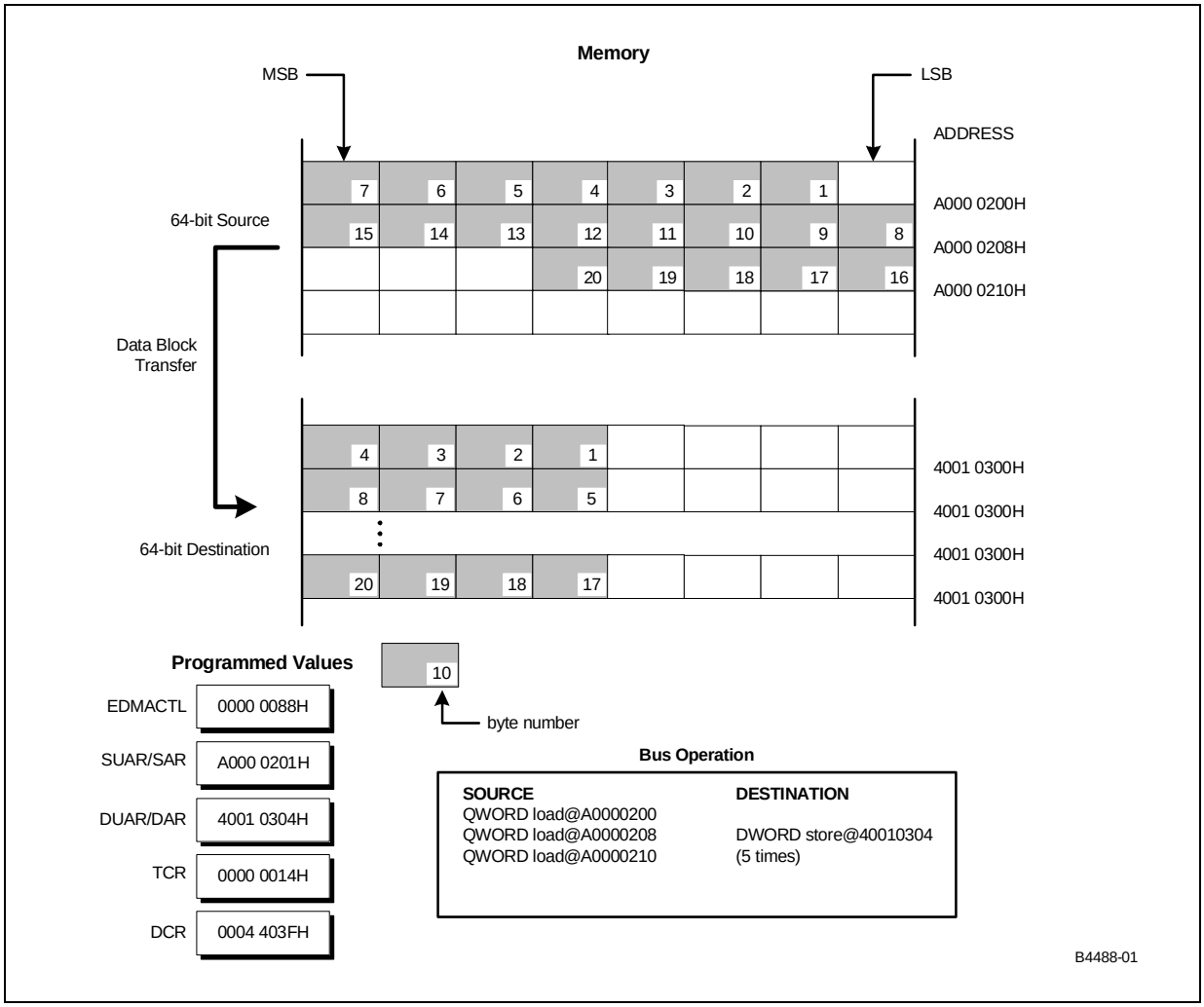


Figure 26. Source in Increment and Destination in 4-Byte Granularity Constant Mode Transfer



The Figure 27, Figure 28, and Figure 29 illustrate data transfers between an unaligned 64-bit, source and a constant destination addresses when the source is in decrement mode.

Figure 27. Source in Decrement and Destination in 1-Byte Granularity Constant Mode Transfer

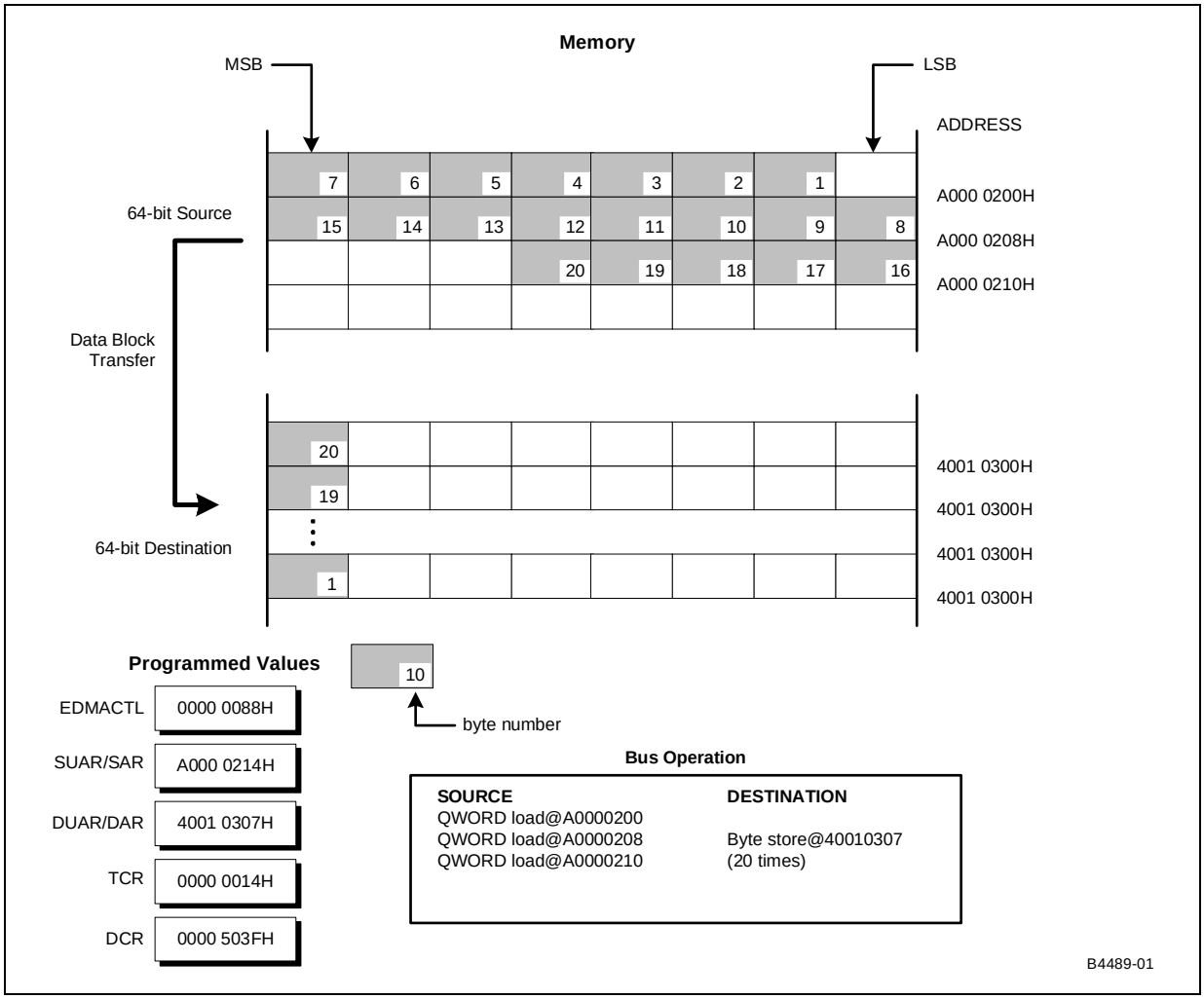


Figure 28. Source in Decrement and Destination in 2-Byte Granularity Constant Mode Transfer

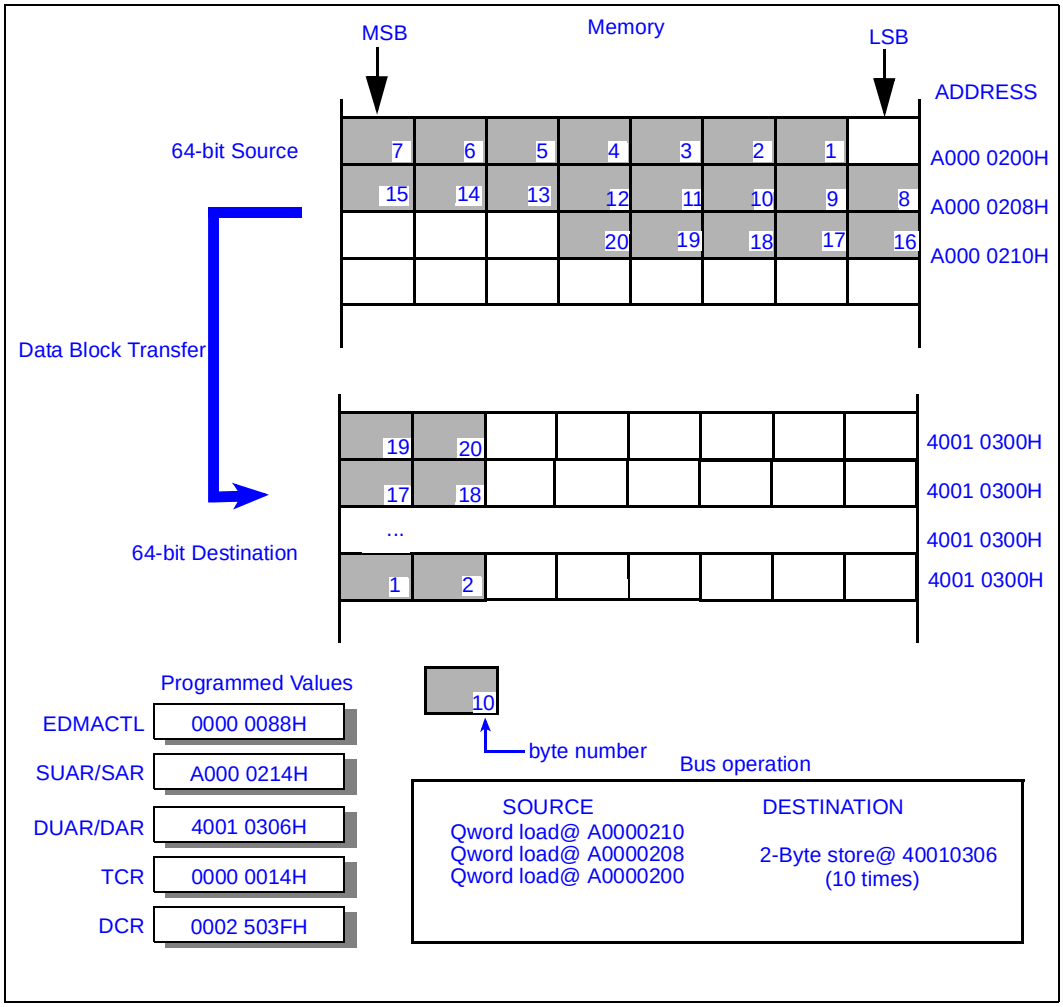
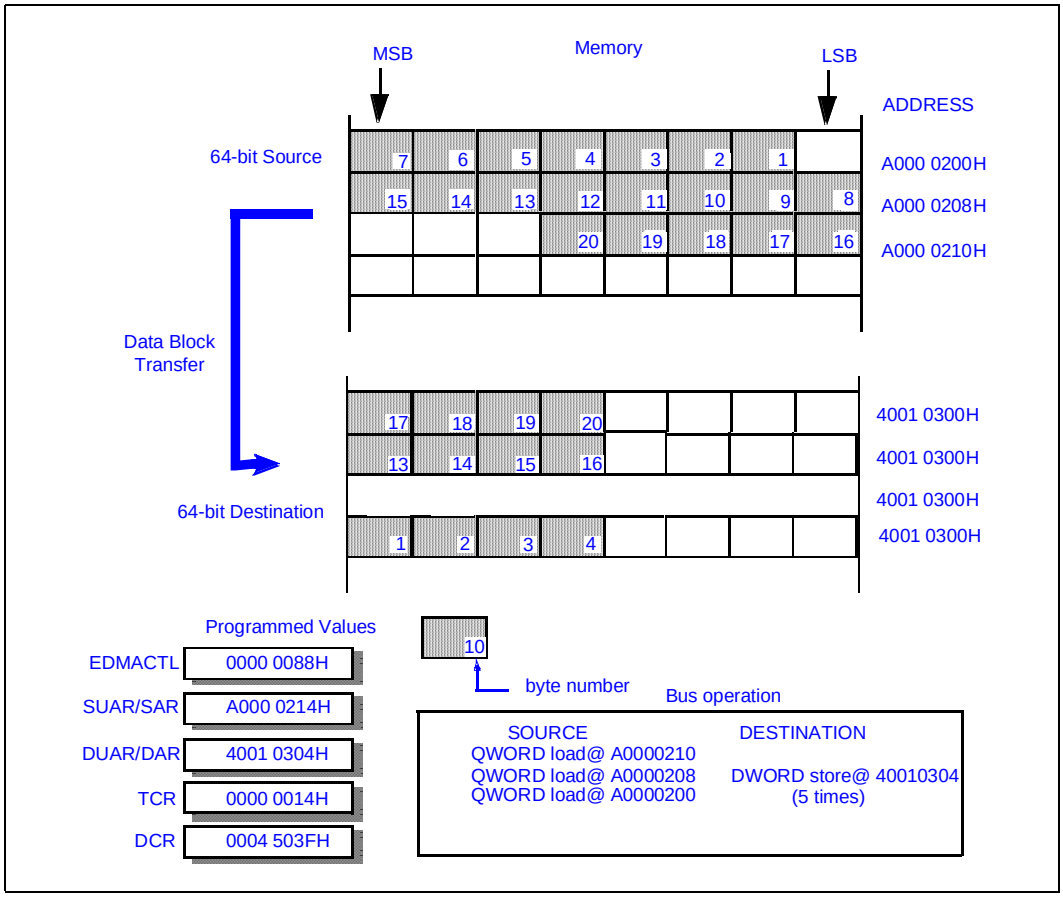


Figure 29. Source in Decrement and Destination in 4-Byte Granularity Constant Mode Transfer



3.5.2.4 Buffer and Memory Initialization Modes

The EDMA can be used to write a constant value to local memory or to memory mapped I/O. As with normal transfers, descriptors are used to specify the memory blocks to which the data contained in the Source Address Register is written.

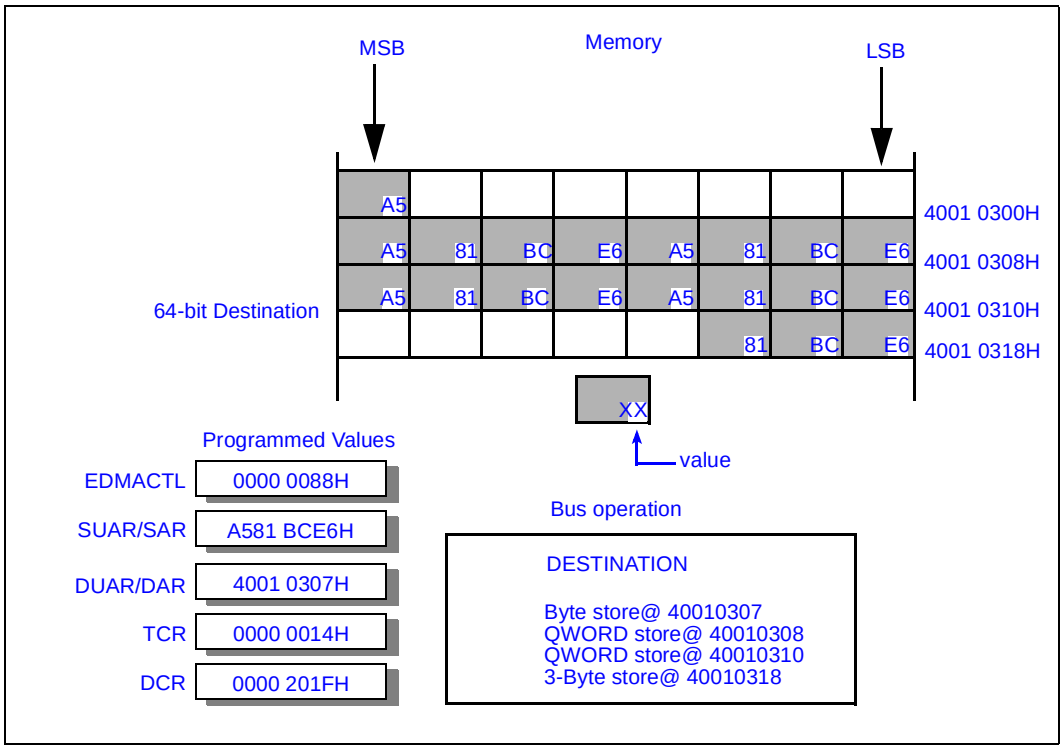
When Memory or Buffer Initialization Modes are selected, the data in the SAR is sent to the destination address. No data is fetched. Data is transferred in 32-bit replicated chunks to the destination. The transfers will continue until the byte count register is satisfied.

3.5.2.4.1 Memory Initialization Mode

Memory Initialization Mode transfers can be to memory or to memory mapped I/O. In this mode, the destination can be specified down to the byte address.

Figure 30 illustrates Memory Initialization to an arbitrary destination address.

Figure 30. Source in Memory Initialization and Destination in Increment Mode Transfer



3.5.2.4.2 Buffer Initialization Mode

Buffer Initialization mode transfers will logically only be to memory mapped I/O and utilize the constant destination mode and granularity fields of the DCR. The address granularity is dictated by the granularity field in the DCR. No errors will be flagged if the destination address is not matched to the granularity but the required lower address bits will be ignored.

Figure 31 though Figure 33 illustrate Buffer Initialization Mode to an arbitrary destination address.

Figure 31. Source in Buffer Init and Destination in 1-Byte Granularity Constant Mode Transfer

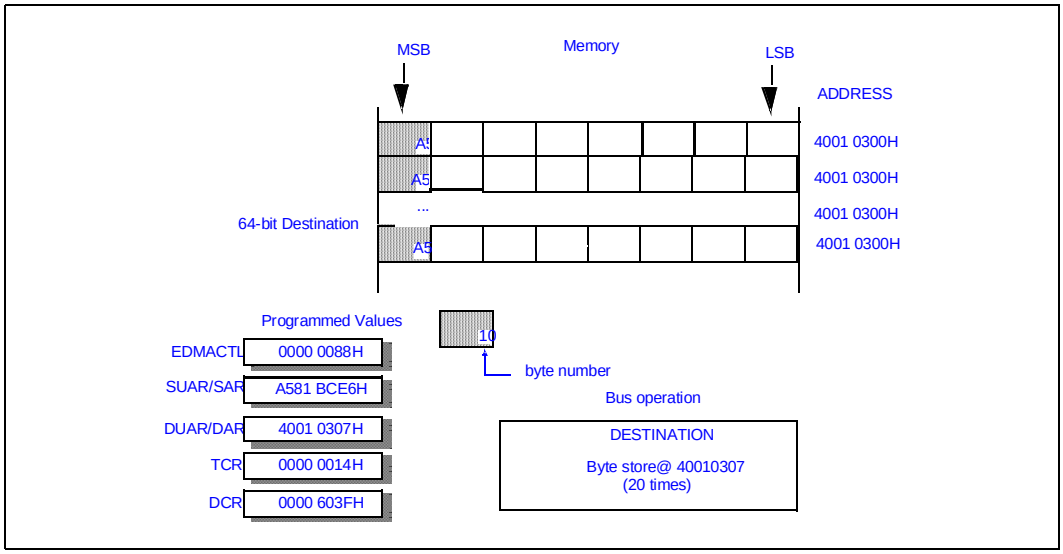


Figure 32. Source in Buffer Init and Destination in 2-Byte Granularity Constant Mode Transfer

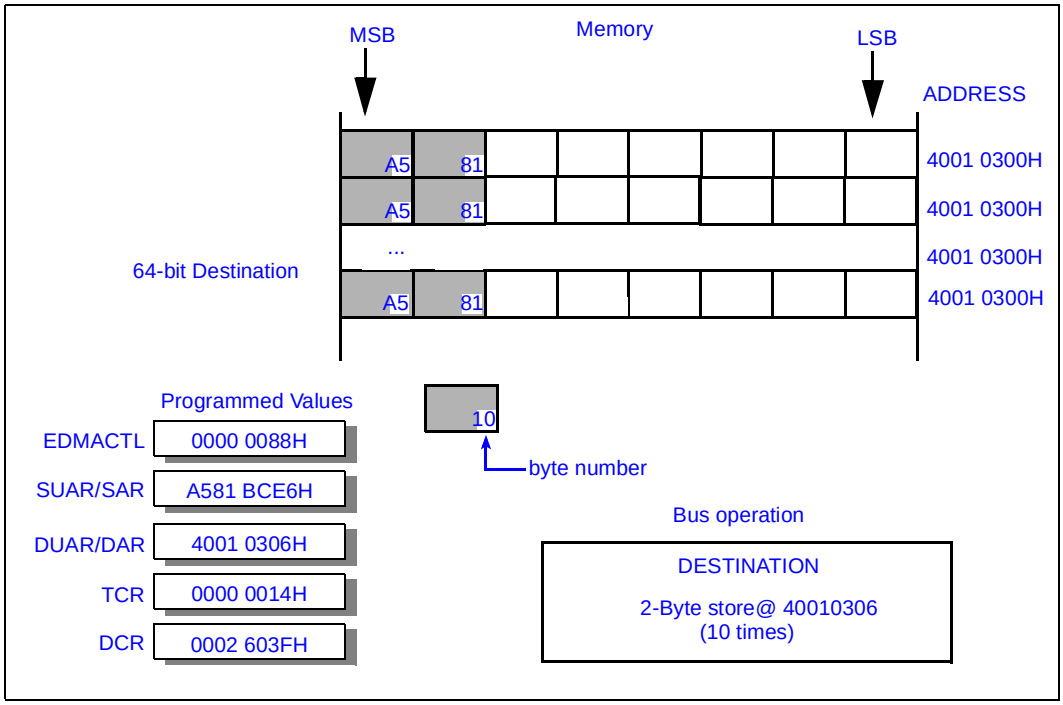
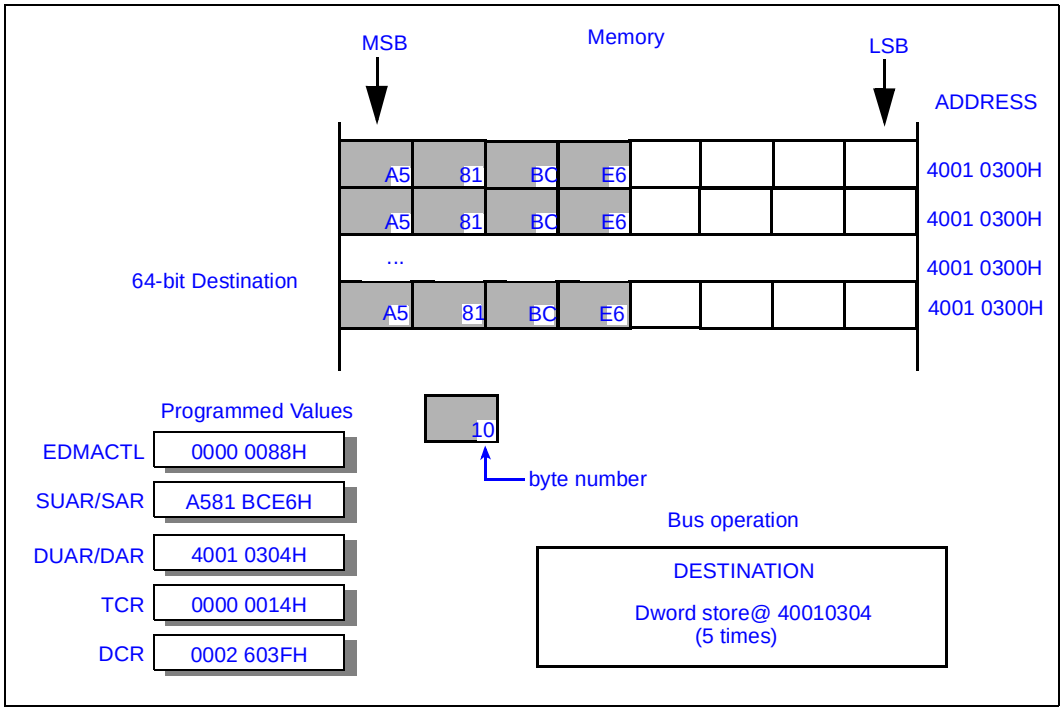


Figure 33. Source in Buffer Init and Destination in 4-Byte Granularity Constant Mode Transfer



3.5.3 PCI Express A Port Traffic Class

To provide traffic shaping and quality of service within the system fabric, the EDMA contains the traffic class field within the DCR. Traffic class is provided to the external switches, bridges, and other fabric devices to allow one transaction to pass another based on a software generated priority map. The three bit field in the DCR is directly copied to the transaction traffic class field in the PCI Express header as write transactions are generated to the PCI Express A port. This field is ignored for all other destinations. PCI Express B port transactions are not supported.

3.6 Channel Data Queuing

Each channel contains a data buffer that is four cache-lines in size (256 bytes). The data buffer holds data temporarily to facilitate pipelining and hide latency, improving the throughput of data transfers between the source and destination.

3.7 Error Conditions

Any of several possible error conditions may arise during a transfer depending on which interfaces the transfer utilizes. The interfaces covered are the EDMA controller interface, the memory interface, and the I/O subsystem destination port interface. All error conditions are reported by setting the corresponding error bits in the Channel Status Register (CSR).

The sections below describe all possible errors at each interface that the EDMA controller must detect and report. For those errors resulting in a channel abort, the response to the error is highly configurable. The controller may be configured to

generate a processor interrupt upon detection of an error. Above and beyond this mechanism, errors detected and logged in the CSR may be escalated as described in the chapter on RASUM and exception handling.

Refer to the EDMA access disposition tables in Chapter 5.0, “System Address Map,” for an overview of defined ranges in the memory map and associated EDMA access treatment.

3.7.1 Controller Interface Error

The following errors may be reported for any EDMA initiated access, regardless of target interface.

- Illegal NDAR address
  - — The descriptor pointer in NDAR is not naturally 8 Dword aligned
  - — The value in NDAR does not point to a valid memory location
- Illegal source address
  - — Address does not comply with the Source Type bit in the DCR
  - — Address out of range, (non-zero bits detected above bit 35 in the descriptor)
- Illegal destination address
  - — Address does not comply with the Destination Type bit in the DCR
  - — Address out of range, (non-zero bits detected above bit 35 in the descriptor)
  - — Accesses to the PCI Express B port or any IICH address region via the NSI
- Received Configuration/TAP write to protected registers
- Data parity error (poisoned data) returned by memory read retrieving descriptor information

All controller interface errors are fatal to the transfer in process and will result in channel abort.

Note: This includes data parity errors, since an error in reading a descriptor implies a corrupt descriptor in main memory, and in this case, it is impossible for each channel to determine precisely what part of the descriptor is damaged. The automatic abort upon detection of a corrupt descriptor is necessary to prevent any further data corruption as a result of its execution.

3.7.2 Memory Interface Error

The following errors may be reported for an EDMA initiated access (read or write) on the local memory interface.

- Addressing error (source or destination)
  - — Physical EDMA address above REMAPLIMIT (see Section 5.6.1)
  - — Physical address not allocated to memory (including PAM destination mapping)
  - — Physical address specified an illegal memory destination (e.g., protected SMM range)
- Data parity error in reading data from the EDMA Data Queue
- Data parity error (poisoned data) returned by memory read for payload data

There is no time-out mechanism associated with transfers. The EDMA channel assumes that all reads to memory will eventually return, although they may return poisoned data, and will wait indefinitely for an outstanding read.

Addressing errors are fatal and will result in a EDMA channel abort, logged in CSR bit 4. The channel response to data errors is configurable; the channel may be programmed to abort, or to propagate the bad data to its destination.

3.7.3 I/O Interface Error

The following errors may be reported for a EDMA initiated access (write) on an I/O interface.

- Address crossed into a memory destination range (checked on each write access)
- Address crossed to a new destination port during a transfer (checked at 4 Kbyte boundaries)

The latter error will result if poorly formed destination descriptor information specifies a length plus address combination that crosses the addressing boundary between independent outbound ports on the I/O subsystem. Any transfer with a destination range crossing an aligned 4 Kbyte boundary in address space may encounter this error.

Addressing errors are fatal and will result in a channel abort. Channel response to data errors is configurable; each channel may be programmed to abort, or to propagate the bad data to its destination.

3.8 Channel Arbitration

Arbitration among the four independent channels occurs in two stages. Each channel has an independent bus request/grant pair to the arbiter internal to the controller. The controller in turn has a single request/grant pair to the main arbiter. The arbiter within the controller handles the fairness among channels, while the inbound/outbound arbiter handles fairness between the EDMA channels and other competing traffic sources.

The internal arbiter uses a strict round robin policy, with the added modification of an optional “high priority” designation for one channel at any given time. Thus a set of competing channels will achieve balanced bandwidth performance during normal operation.

The inbound/outbound arbiter provides a programmable single or double “grant duration” for the EDMA controller. Thus the channel that “wins” internal arbitration may be allowed to issue one or two access requests back-to-back in a single arbitration cycle. The second request is accepted if the inbound/outbound arbiter is programmed to a grant count of two and the requesting channel has two consecutive requests of the same type targeting the same destination ready to send, and there are sufficient command and data resources available for the second request.

3.8.1 Normal Arbitration Scheme

A fully connected round-robin arbiter provides a distinctive balanced service among competing requestors. Each of the actively competing channels will receive an equal fraction of the bandwidth service provided by the inbound/outbound arbiter on behalf of each EDMA channel. In the absence of any competition from the processor, PCI Express ports, or other I/O, each EDMA channel will be allowed to saturate the memory interface. For example, given a memory interface “saturation point” of 4 GB/s, the round-robin scheme would be equally distributed between the competing EDMA channels.

3.8.2 Prioritized Arbitration Scheme

The high priority option in the arbitration scheme provides for a single “high priority” channel to receive favorable latency and bandwidth service in the face of multiple competing “normal priority” channels. This is accomplished by designating the high priority channel using a priority enable bit and a two-bit field (to select one of four channels) in the EDMA control register.

The internal EDMA controller arbiter modifies its arbitration algorithm to provide a grant to the priority channel between the grants for each of the other channels, which retain their round-robin prioritization relative to each other. For example, given a hypothetical memory interface bandwidth of 4 GB/s, the priority-modified scheme would result in 2 GB/s (half the available bandwidth) for the designated priority channel, and the remaining 2 GB/s split equally among the other competing channels.

The limitation that a single channel at a time be designated to be the priority channel is an acknowledgement that quality of service differences given multiple “priority” channels would be slight in this implementation with two-level arbitration and only four competing channels. It is anticipated that software will be able to determine when an application is particularly sensitive to service level, has been allocated a channel, and will manage the assignment of priority accordingly. If more than one such “sensitive” application is in flight at the same time, it is perceived to be more effective to allow fair competition between those sources, and let kernel or device driver software attempt to manage competition for resources at the system level to prevent service level problems.

Each EDMA channel supports modification to the priority channel settings on the fly (while one or more channels are active). A write to the control register that changes the priority channel configuration takes effect at the next arbitration decision point after the write has completed. There is no direct interlock between the arbiter configuration and any of the active channels. Such an event is effectively an environment change orthogonal to work in progress on any given channel.

3.9 Configuration

The EDMA controller uses memory-mapped configuration registers for the majority of its per channel register sets. The controller is software compatible with standard PCI device configuration, and implements a standard PCI header in its configuration-mapped register set. The memory-mapped register space associated with the controller is identified by a 32-bit memory Base Address Register (BAR). Table 57 provides an overview of the memory-mapped register set for a representative channel of the controller.

Table 57. Channel 0 Memory-mapped Register Set

Memory Mapped I/O for EDMA Channel 0	Memory offset	Access	Size	Default	Sticky
Channel Control Register (CCR0)	00-3h	RW	32 bits	0000_0000h	No
Channel Status Register (CSR0)	04-07h	RWC, RO	32 bits	0000_0000h	No
Current Descriptor Addr Reg (CDAR0)	08-0Bh	RO	32 bits	0000_0000h	No
Current Descriptor Upper Addr Reg (CDUAR0)	0C-0Fh	RO	32 bits	0000_0000h	No
Source Address Register (SAR0)	10-13h	RO	32 bits	0000_0000h	No
Source Upper Address Register (SUAR0)	14-17h	RO	32 bits	0000_0000h	No
Destination Address Register (DAR0)	18-1Bh	RO	32 bits	0000_0000h	No
Destination Upper Address Register (DUAR0)	1C-1Fh	RO	32 bits	0000_0000h	No
Next Descriptor Address Register (NDAR0)	20-23h	RWL	32 bits	0000_0000h	No
Next Descriptor Upper Address Register (NDUAR0)	24-27h	RWL	32 bits	0000_0000h	No
Transfer Count Register (TCR0)	28-2Bh	RO	32 bits	0000_0000h	No
Descriptor Control Register (DCR0)	2C-2Fh	RO	32 bits	0000_0000h	No

All the internal registers are accessible through host-initiated configuration space accesses, and through the TAP or SMBus interface accesses. Internal registers are not accessible from the I/O subsystem interfaces. See Figure 4, "Device 1, Function 0 EDMA Registers" on page 66 for the EDMA configuration register map.

3.9.1 Power Up/Default Status

Upon power-up or hardware reset, the channel registers are initialized to their default values. All reserved and unimplemented registers and bits in the device return zero on reads and are unaffected by writes.

3.9.2 Channel-Specific Register Definitions

Each channel has twelve 32-bit memory-mapped registers for its independent operation. Eight of these registers (refer to descriptions below) are loaded automatically from their corresponding fields in the chain descriptor when a new descriptor is fetched from local memory during normal operation. The format of the corresponding descriptor fields in memory is identical to the format defined for the channel-specific registers. Refer to "Memory Mapped I/O for EDMA Registers" on page 511 for bit definitions.

Read/write access is available only to the:

- Channel Control Register (CCR)
- Channel Status Register (CSR)
- Next Descriptor Address Register (NDAR)
- Next Descriptor Upper Address Register (NDUAR)

The remaining registers are read-only and are automatically loaded with new values defined by the chain descriptor whenever the channel reads a chain descriptor from local system memory.

Note: Automatic loading of the channel-specific registers occurs after the memory read completion returns the descriptor data (32 Byte), and verification has taken place. Verification includes checking parity on the data returned, and checking that the channel is properly configured to receive new descriptor data. (If a suspend is in progress, the descriptor data will be dropped in honor of the suspend.)

3.9.2.1 Channel Control Register – CCR

The Channel Control Register (CCR) specifies the overall operating environment for the channel. This is a Read/Write register, and is cleared to zero on power-on or system reset (contains no sticky bits). Application software initializes this register only after initializing the chain descriptors in system memory and updating the Next Address Registers with the location of the first chain descriptor in memory. The CCR may be written when the channel is active to modify channel operation (stop, suspend, etc.) while the channel is active.

The following bits are defined in the CCR:

- Start: initiate a new transfer (requires that the CSR be appropriately cleared)
- Stop: abort the current transfer (immediately)
- Suspend: suspend the current transfer (upon completion of the current descriptor)
- Channel Resume: resume a suspended transfer (retrieve the descriptor indicated by NDAR/NDUAR from local memory, and proceed with execution per the value returned), (requires that the stopped and abort status bits in the CSR be clear to take effect, will automatically clear end of chain and end of transfer flags)

Refer to "Offset 00 - 03h: CCR0 – Channel 0 Channel Control Register" on page 512 for the format of the CCR.

3.9.2.2 Channel Status Register – CSR

Channel Status Register (CSR) contains flags to indicate the channel status. The register is read by application software to get the current channel status and to determine the source of interrupts. CSR is cleared to zero on power-on or system reset. This is a Read/Write register.

The following bits are defined in CSR:

- Channel Active: transfer in progress
- Aborted: transfer encountered an error
- Stopped: transfer stopped via software request (Stop bit detected)
- Suspended: transfer suspended via software request (Suspend bit detected)
- End of Transfer: channel has completed execution of (at least one) descriptor
- End of Chain: channel has completed execution of the terminal descriptor (null NDAR/NDUAR)

Refer to "Offset 04 - 07h: CSR0 – Channel 0 Channel Status Register" on page 514 for the format of CSR.

3.9.2.3 Current Descriptor Address Register – CDAR

The Current Descriptor Address Register (CDAR) contains the lower 32-bits of the address of the current chain descriptor in local system memory. The CDAR is cleared to zero on power-on or system reset, and is loaded automatically with the value from the Next Descriptor Address Register (NDAR) when a new block transfer is initiated. This register is read-only, and may be polled by software to monitor the progress of the channel as it traverses the descriptor chain.[1]

3.9.2.4 Current Descriptor Upper Address Register – CDUAR

The Current Descriptor Upper Address Register (CDUAR) contains the upper 32-bits of the address of the current chain descriptor in local system memory. The CDUAR is cleared to zero on power-on or system reset, and is loaded automatically with the value from the Next Descriptor Upper Address Register (NDUAR) when a new block transfer is initiated. This register is read-only[2]

3.9.2.5 Source Address Register – SAR

The Source Address Register (SAR) contains the lower 32-bits of the source address for the current transfer. The SAR is cleared to zero on power-on or system reset, and is loaded automatically with the Source Address field of the chain descriptor (first DWord) when a new chain descriptor is read from memory. The address can be aligned to any byte boundary. The system destination for reads to this address range must match the Source Type setting of the DCR or the transfer will abort.

---

1. Note that the IMCH does not provide an interlock to guarantee that consecutive reads to the CDAR/CDUAR pair return portions of the same descriptor in the event of a collision between the read accesses and a descriptor load operation. If software requires knowledge of the current descriptor, the "Suspend" function must be invoked prior to polling these registers.
2. Note that the IMCH does not provide an interlock to guarantee that consecutive reads to the CDAR/CDUAR pair return portions of the same descriptor in the event of a collision between the read accesses and a descriptor load operation. If software requires knowledge of the current descriptor, the "Suspend" function must be invoked prior to polling these registers.

3.9.2.6 Source Upper Address Register – SUAR

The Source Upper Address Register (SUAR) contains the upper 32-bits of the source address for the current transfer. The SUAR is cleared to zero on power-on or system reset, and is loaded automatically with the Source Upper Address field of the chain descriptor (second DWord) when a new chain descriptor is read from memory. All address bits above bit 35 must be zero or the transfer will abort and an error will be reported.

3.9.2.7 Destination Address Register – DAR

The Destination Address Register (DAR) contains the lower 32-bits of the destination address for the current transfer. The DAR is cleared to zero on power-on or system reset, and is loaded automatically with the Destination Address field of the chain descriptor (third DWord) when a new chain descriptor is read from memory. The address can be aligned to any byte boundary. The system destination for writes to this address range must match the Destination Type setting of the DCR or the transfer will abort.

3.9.2.8 Destination Upper Address Register – DUAR

The Destination Upper Address Register (DUAR) contains the upper 32-bits of the destination address for the current transfer. The DUAR is cleared to zero on power-on or system reset, and is loaded automatically with the Destination Upper Address field of the chain descriptor (fourth DWord) when a new chain descriptor is read from memory. All address bits above bit 35 must be zero or the transfer will abort and an error will be reported.

3.9.2.9 Next Descriptor Address Register – NDAR

The Next Descriptor Address Register (NDAR) contains the lower 32-bit address of the next descriptor chain in the local system memory. The NDAR is cleared to zero on power-on or system reset, and is loaded automatically with the Next Descriptor Address field of the chain descriptor (fifth DWord) when a new chain descriptor is read from memory. This address must be aligned to an 8-DWord address boundary. A value of zero implies the end of chain if the value of Next Descriptor Upper Address (loaded into the NDUAR) is also zero. Application software writes this register with the address of the first chain descriptor in memory prior to initiating a transfer.

Note: The application software must make sure that the Start bit in the CCR and the Channel Active bit in the CSR are clear prior to writing to the NDAR. The IMCH protects this register from being written when these bits are not clear. If the NDAR and NDUAR are zero when the Start bit is set, no transfer will be initiated.

3.9.2.10 Next Descriptor Upper Address Register – NDUAR

The Next Descriptor Upper Address Register (NDUAR) contains the upper 32-bit address of the next descriptor chain in the local system memory. All address bits above bit 35 must be zero or the transfer will abort and an error will be reported. A value of zero implies the end of chain if the value of Next Descriptor Address (loaded into the NDAR) is also zero. NDUAR is cleared to zero on power-on or system reset, and is loaded automatically with the Next Descriptor Upper Address field of the chain descriptor (sixth DWord) when a new chain descriptor is read from memory. Application software (likely the device driver) writes this register with the address of the first chain descriptor in the memory prior to initiating a transfer.

Note: The application software must make sure that the Start bit in the CCR and the Channel Active bit in the CSR are clear prior to writing to the NDUAR. The IMCH protects this

register from being written when these bits are not clear. If the NDAR and NDUAR are zero when the Start bit is set, no transfer will be initiated.

3.9.2.11 Transfer Count Register – TCR

The Transfer Count Register (TCR) contains the length of the current transfer in bytes. The TCR is cleared to zero on power-on or system reset, and is loaded automatically with the Transfer Count field of the chain descriptor (seventh DWord) when a new chain descriptor is read from memory. The TCR allows for a maximum transfer of 16 Mbytes, commensurate with current operating system capabilities. A value of zero is valid and results in no data being transferred, and no cycles generated on the source or destination buses.

3.9.2.12 Descriptor Control Register – DCR

The Descriptor Control Register (DCR) contains control values for the transfer on a per descriptor basis. The DCR is cleared to zero on power-on or system reset, and is loaded automatically with the Descriptor Control field of the chain descriptor (eighth DWord) when a new chain descriptor is read from memory. The values in the DCR may vary for different descriptors within a single chain.

Note: The descriptor control register value stipulates coherence attributes for both the source and destination addresses defined by this chain descriptor. Independent bits are also defined to specify whether the source and destination address ranges are to be treated as “coherent” or “non-coherent” by the IMCH. When the DCR value stipulates that one or both of the source and destination are to be treated as “non-coherent” space, the IMCH will rely on software to maintain system memory coherency, and will not issue FSB cycles during the block transfer to snoop processor caches on behalf of the corresponding address range(s).

The following bits are defined in the CCR:

- Destination Address Mode: two bits specify destination address as increment or constant
- Granularity of the transfer in destination constant address mode: two bits (1B, 2B, or 4B)
- PCI Express A port Destination Traffic Class: three bits define this traffic class
- Source Address Mode: two bits specify source address as increment, decrement, or buffer/memory initialization
- Buffer/Memory Initialization Mode: Specifies a write to fill an area of memory
- Destination Coherency: specifies whether destination addresses should be snooped on the FSB
- Source Coherency: specifies whether source addresses should be snooped on the FSB
- Destination Type: specifies whether the destination is local memory or the I/O subsystem
- Source Type: specifies whether the source is local memory or the I/O subsystem (defined only for symmetry, I/O subsystem source addresses are not supported)
- Abort Interrupt Enable: specifies whether to generate an interrupt on abort
- Stop Interrupt Enable: specifies whether to generate an interrupt on stop
- Suspend Interrupt Enable: specifies whether to generate an interrupt on suspend
- End of Transfer Interrupt Enable: specifies whether to generate an interrupt on EOT
- End of Chain Interrupt Enable: specifies whether to generate an interrupt on EOC

Refer to Chapter , “Offset 2C - 2Fh: DCR0 – Channel 0 Descriptor Control Register,” for the format of the DCR.

3.10 Interrupts

Each EDMA channel can be configured to generate interrupts to the processor interface. The interrupt enable bits for end of transfer and end of chain in the Descriptor Control Register (DCR) determine if the channel generates an interrupt upon successful error-free completion of a transfer. The Abort Interrupt Enable bit in the DCR determines if the channel generates an interrupt upon encountering an error. Refer to “Error Conditions” on page 132 for details on errors on both the source and destination interface. Table 58 summarizes the status flags, and the conditions under which interrupts will be generated.

Each chain descriptor can independently set or clear the various interrupt enable bits in the Descriptor Control Register. This level of control for interrupt generation permits flexibility in synchronization between application software and transfers in progress. If interrupts are not enabled, synchronization can be achieved by polling the status bits in the Channel Status Register (CSR).

Table 58. Interrupt Summary

Interrupt Conditions	Channel Status Register (CSR) Flags						DCR Bit Settings (INTR Enable Bits)				
	Channel Active	Stopped	Suspended	End of Transfer	End of Chain	Channel Aborted (Error)	Stop INTR Enable	Suspend INTR Enable	EOT INTR Enable	EOC INTR Enable	Abort INTR Enable
Stopped	0	1	0[2]	-	-	-	1[4]	-	-	-	-
Suspended	0	0[2]	1	1[3]	-	-	-	1	-	-	-
End of Transfer	1	-	-	1	-	-	-	-	1	-	-
End of Chain	0	-	-	1	1	-	-	-	-	1	-
Channel Abort[1]	0	-	-	0	0	1	-	-	-	-	1

Notes:

1. The IMCH ensures that any aborted transfer will be reported via the Channel Abort status bit and that this bit will never be accompanied by an End of Transfer or End of Chain indication. This ensures that software never mistakes an aborted transfer for a successfully completed transfer – even if the error is not detected until the final write to the final destination address of the terminal chain descriptor.
2. The Stop and Suspend functions are mutually exclusive, and only one of the two status bits will ever be asserted by the IMCH. In the event that software asserts both controls in the CCR, the Stop function will take precedence.
3. The EDMA Suspend function causes the channel to suspend operation at the completion of the current descriptor. The EOT status bit will always accompany the suspended status bit. Note that even if interrupts are enabled for both EOT and suspend, only a single interrupt event will result.
4. The Stop function causes the channel to abort the transfer in progress immediately. It is recommended that software read back the channel status register to verify that a stop command has taken effect, since this will be much faster than setting the interrupt enable for stop and waiting for the interrupt to occur.

3.10.1 Interrupt Routing Mechanisms

Two different mechanisms are available to route interrupts generated by channels to the processor. Note that the interrupt mechanism itself is not channel-specific; all channels generating interrupts share the same interrupt vector and handler. This is in line with the expectation that a single device driver controls each EDMA channel at large, rather than independent drivers per channel.

The first interrupt mechanism uses the integrated IOxAPIC and 8259 emulation hardware. All interrupts from the channels are logically ORed and routed to the interface controller for propagation via the in-band Assert_Intx and Deassert_Intx special cycles, emulating a level-sensitive interrupt output. The IMCH tracks these special cycles, and forwards the signaled interrupt to the processor. If the APIC enable bit is set, an interrupt will result in an APIC message. If the APIC enable bit is clear (unanticipated but possible), an interrupt will result in a legacy mode 8259 style level sensitive interrupt directly to the processor socket.

The second interrupt mechanism uses Message Signaled Interrupt (MSI) generation functionality integrated into the EDMA. Internal interrupt messaging utilizes the PCI message capability structure, and does not support external interrupt input routing.

That is, the limited MSI functionality described here is dedicated to the EDMA channel. This second mechanism is preferred for interrupt signaling, but is only available in platforms running an MSI-capable operating system.

Selection between these two mechanisms is automatic in the IMCH. If MSI messaging is enabled as indicated by the enable bit in the MSI control register, then the MSI interrupt mechanism is used. If MSI message generation is disabled, any initiated interrupt will use the dedicated pin legacy mechanism.

For both interrupt mechanisms, the interrupt service routine (ISR) must service all interrupts for all channels. The memory-mapped EDMA Controller Global Status Register must be statused before returning from the ISR to ensure no additional interrupts have occurred. Failure to address interrupts in all channels will result in potential system starvation.

3.10.2 Message Signaled Interrupt (MSI)

The EDMA controller is capable of generating upstream interrupt messages (MSI) directly to the processor. An MSI is signaled via a Memory Write to address 0FEEx_xxxxh. Refer to Section 7.4, "Interrupt Delivery" on page 277 for details.

Three 32-bit registers are required in the controller to support this mechanism. The default values of these registers are compatible with the default value of the IOxAPIC specification. The three registers are the MSI Control Register (MSICR), MSI Address Register (MSIAR), and MSI Data Register (MSIDR). Software must program these registers to appropriate values prior to enabling internal MSI functionality.

Note: It is unsafe to enable the integrated MSI APIC function of the controller in environments under control of a non MSI-capable operating system.

The MSI mechanism supports differentiation between interrupts generated during normal operation (EOT, EOC, stop, and suspend) and interrupts due to errors (abort). This extra level of granularity is unavailable via the legacy interrupt mechanism.

To facilitate use of a single device driver for the entire EDMA function, a single MSI register set services all channels. The support for two different messages on behalf of the controller is included in the MSI register set. Refer to "Device 1, Function 0: EDMA Registers" on page 394 for the format of these registers.

Note: The integrated APIC functionality will not support level-sensitive interrupt emulation requiring the use of broadcast EOI cycles from the FSB. No path is provided to handle such traffic from processor to EDMA control engine; thus, the only supported MSI type is the edge-triggered variety.

The following subsections describe the register set for MSI support.

3.10.2.1 MSI Control Register – MSICR

The MSI Control Register (MSICR) contains control information for MSI interrupt capability. The multiple-message enable field and MSI enable are contained in this register.

3.10.2.2 MSI Address Register – MSIAR

The MSI Address Register (MSIAR) contains address information specifying the message destination address for MSI interrupts.

3.10.2.3 MSI Data Register – MSIDR

The MSI Data Register (MSIDR) contains routing and priority data for generation of MSI interrupts.

3.10.3 Interrupt Ordering

To support MSI signaling as transfers complete, the IMCH must take special steps in hardware to ensure that processor accesses to memory in response to MSI do not experience producer/consumer ordering failures. Specifically, the chip must internally guarantee functionality equivalent to a logical “FENCE” operation between the MSI and subsequent processor traffic from the FSB.

3.10.3.1 Interrupt Ordering for Memory Destination

The failure to be prevented for interrupts signaled at the end of transfers to memory destinations is as follows:

- Each EDMA channel is programmed to move a single cache-line of data and issue an MSI upon completion. This is a special case, for illustrative purposes; a similar scenario arises for the last few writes of a multi-line transfer.
- As soon as the write data are posted into the inbound/outbound arbiter headed for the memory interface, the MSI is issued directly to the FSB. Note that the transfer to memory may be issued without an accompanying FSB snoop cycle (non-coherent), thus the data and interrupt message are logically traversing independent traffic paths.
- In response to the MSI, the processor issues a memory read to retrieve the data from memory. In the absence of an internal interlock, this read may proceed to the memory controller before the posted write data are accepted into the memory controller from the inbound/outbound arbiter. This is where the error occurs, because the memory controller will not have any information regarding the relative issue order of the write and the read – if the read gets there first, it will retrieve stale data.

To prevent such a failure, the inbound/outbound arbiter includes specialized hardware to guarantee that all posted write data received ahead of an MSI are forwarded out of the arbiter before the MSI message will be forwarded to the FSB.

A similar interlock in the inbound/outbound arbiter prevents failures in the non-interrupt case. When software is polling the Channel Status Register (CSR) to detect transfer completion, specialized hardware guarantees that the read completion stalls until all prior posted write data are forwarded out of the arbiter.

3.10.3.2 Interrupt Ordering for Outbound Destination

The failure to be prevented for interrupts signaled at the end of transfers to outbound port destinations is as follows:

- Each EDMA channel is programmed for a single block transfer to an I/O device, with interrupt notification enabled at the completion of that transfer. For this example, MSI generation is disabled.
- When the final write of the transfer is posted into the inbound/outbound arbiter, a level-sensitive interrupt is signaled directly to the interrupt output pin, bypassing all internal queue structures.
- If the integrated IOxAPIC is enabled, the response to the interrupt pin will be an APIC message received. If the APIC is disabled, a sideband interrupt is signaled directly to the processor via a level sensitive output.

- For either case, the likely software response to the interrupt will be a posted write "door-bell" access to a memory mapped control register in the EDMA destination device to communicate completion of the transfer.
- In the absence of internal interlocks as described above, multiple failures are possible. The APIC message could bypass EDMA data pending within the inbound/outbound arbiter, and the processor doorbell write could do the same. This would result in stale data "executed" in response to the doorbell.

The internal IMCH hardware interlock preventing APIC messages from being forwarding to the FSB while posted data remains pending in the inbound/outbound arbiter. This prevents most of the problematic behavior in this case. If the APIC is disabled, the processor must retrieve the interrupt vector from the 8259 emulator, and the read completion interlock will then guarantee that all posted write data has cleared the arbiter.

Another potential issue is relaxed write ordering within a system agent en-route to the EDMA destination device. If the hardware were to issue EDMA outbound posted writes and processor posted writes with differing stream ID codes, an intermediary component may allow the processor doorbell access to move around posted EDMA data. This would again result in stale data "executed" by the destination device.

The IMCH could solve this problem by issuing an explicit FENCE between the final EDMA write and the interrupt, preventing subsequent processor accesses from reordering en-route to the destination. A simpler (but more limited) solution is to utilize the same stream ID for all outbound traffic regardless of source. This makes transactions initiated by each EDMA channel indistinguishable from those initiated by any of the processor threads. With no stream ID information to determine reordering legality, an intermediary device must necessarily enforce strong ordering for all accesses outbound.

The single initiator ID in concert with the internal interlock for APIC messages and read completions is sufficient to guarantee proper behavior. The implementation guarantees that any flag write or data read to the destination port will necessarily push the EDMA transfer data ahead of it, ensuring correct producer/consumer operation.

3.11 Initiating an EDMA Transfer

The following sections detail the steps the software must take in programming a channel to initiate a transfer (or chain of transfers). The steps covered include channel initialization, transfer start, and suspend or stop. Each channel is designed to have independent control of interrupt enabling and generation, and independent transfer attribute controls. This provides the greatest flexibility to the application program.

3.11.1 Setup and Initiation

Initializing a channel begins with constructing one or more chain descriptors in local system memory. Each chain descriptor takes the form described in Section 3.3.1, "Chain Descriptor Definition" on page 114. Once the descriptor(s) are defined, the following steps are required to initiate a transfer:

- Ensure the EDMA channel is enabled and in Normal Mode by setting the EDMA Enable and Mode bits of the EDMA Control Register.
- The channel must be inactive/idle prior to starting a transfer. This may be verified by reading the Channel Active bit in the Channel Status Register (CSR), which is clear when the channel is inactive/idle.
- Update the Next Descriptor Address Register (NDAR/NDUAR) with the address of the first chain descriptor in local system memory.

- Clear the Channel Status Register (CSR) of any asserted error or status bits. Each EDMA channel will not initiate a new transfer when an error condition remains in the CSR.
- Clear the Suspend bit and set the Start bit in the Channel Control Register (CCR). Since this is the start of a new transfer and not the resumption of a previous transfer, the Channel Resume bit in the CCR must be clear. (Resume overrides start.)
- The channel starts the transfer by fetching the chain descriptor at the address contained in the Next Descriptor Address Register (NDAR/NDUAR). The channel moves the NDAR/NDUAR values into the CDAR/CDUAR, and loads the chain descriptor values into their corresponding internal registers. If the load completes without any error, the actual data transfer begins. The Current Descriptor Address Register (CDAR/CDUAR) now contains the address of the chain descriptor just fetched, and the Next Descriptor Address Register (NDAR/NDUAR) now contains the descriptor address of the next descriptor in the chain, if any.
- When the current EDMA transfer has completed without any errors, the channel fetches the next chain descriptor from the address contained in the Next Descriptor Address Register (NDAR/NDUAR) automatically without any software intervention, and proceeds with the next block transfer (provided the value in the NDAR/NDUAR pair is non-zero).

The last descriptor in the chain list has a null value in the Next Descriptor Address field, specifying the end of the chain. The null value in the Next Descriptor Address Register (NDAR/NDUAR) notifies the channel not to read additional chain descriptors from local system memory, and the channel goes idle.

3.11.2 Suspend Function

Software may temporarily suspend execution of a descriptor chain by setting the Suspend bit in the Channel Control Register (CCR). The target channel will complete execution of the current descriptor, and suspend operation without losing current status. Software may later cause the channel to resume execution of the descriptor chain by writing to the CCR to clear the Suspend bit and set the Resume bit. In response, the channel will initiate a descriptor fetch from the NDAR/NDUAR, and resume the suspended operation.

It is not necessary for software to reprogram the channel configuration after a suspend sequence.

3.11.3 Stop Function

Software may intentionally abort a transfer by setting the Stop bit in the Channel Control Register (CSR). Once aborted, the transfer cannot be resumed. In response to the Stop bit, the target channel will immediately cease fetching source data, drain any buffered destination data, and go idle. Usage of this mechanism will result in assertion of the Stopped status bit in the CSR, and will generate an interrupt if so enabled. (Note that the stop function is sufficiently fast in the IMCH that reading back the channel status register is preferred over utilizing the interrupt on stop function.)

Usage of the Stop mechanism will not result in assertion of the Abort status bit, nor will it generate an “interrupt on abort” indication if so enabled. The channel differentiates an abort on error from an abort on software command, and will ensure that all error status bits remain clear.

If the MSI mechanism is in use for interrupt generation, and independent messages are defined for abort on error and normal run-time interrupts, the latter message type will be utilized on behalf of the stop function.

3.11.4 EDMA Process Flow

Figure 34 provides a high-level flow chart of the EDMA initialization sequence, and Figure 35 provides a similar view of the EDMA completion sequence.

Figure 34. Initiation Flow Chart

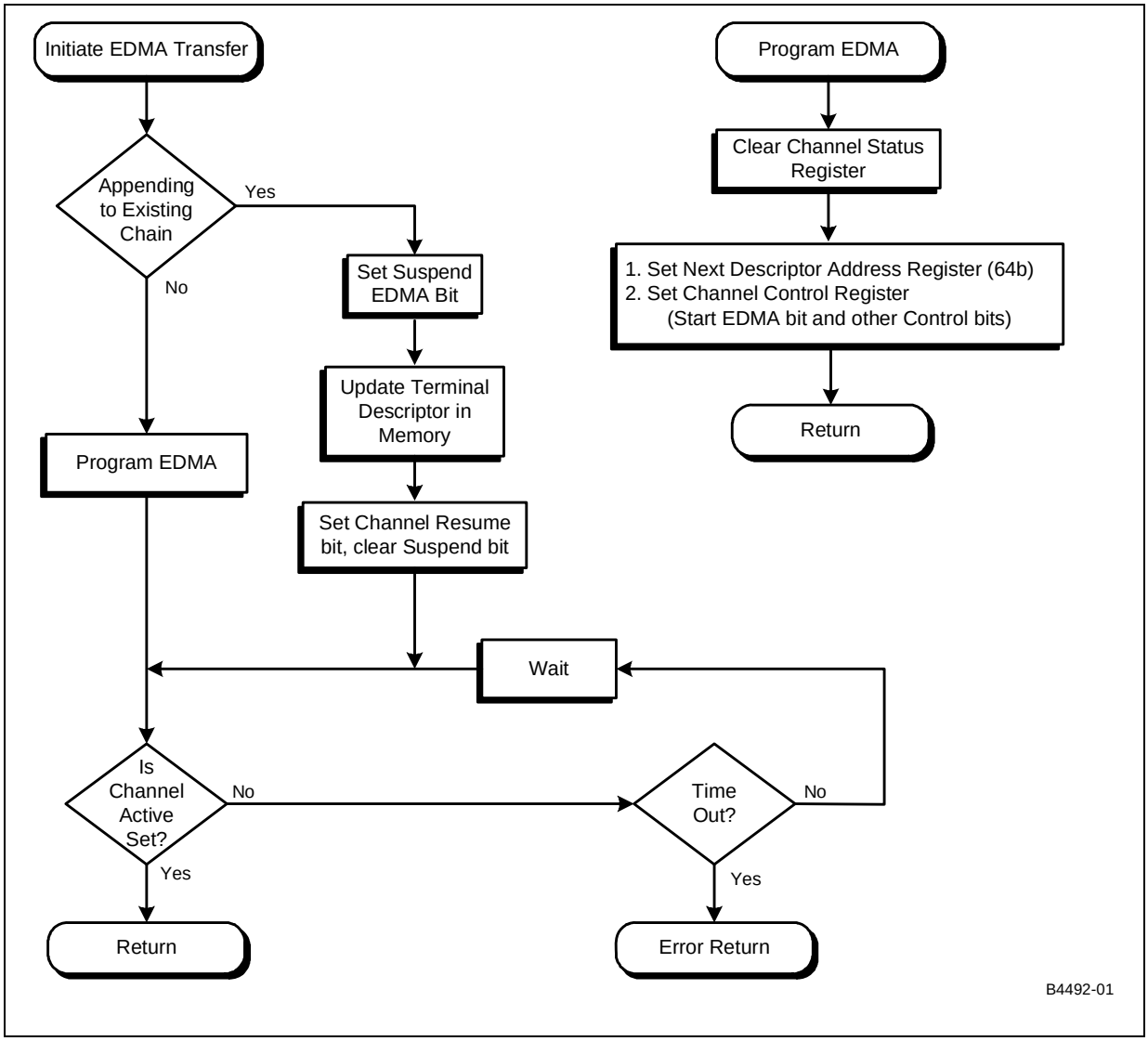
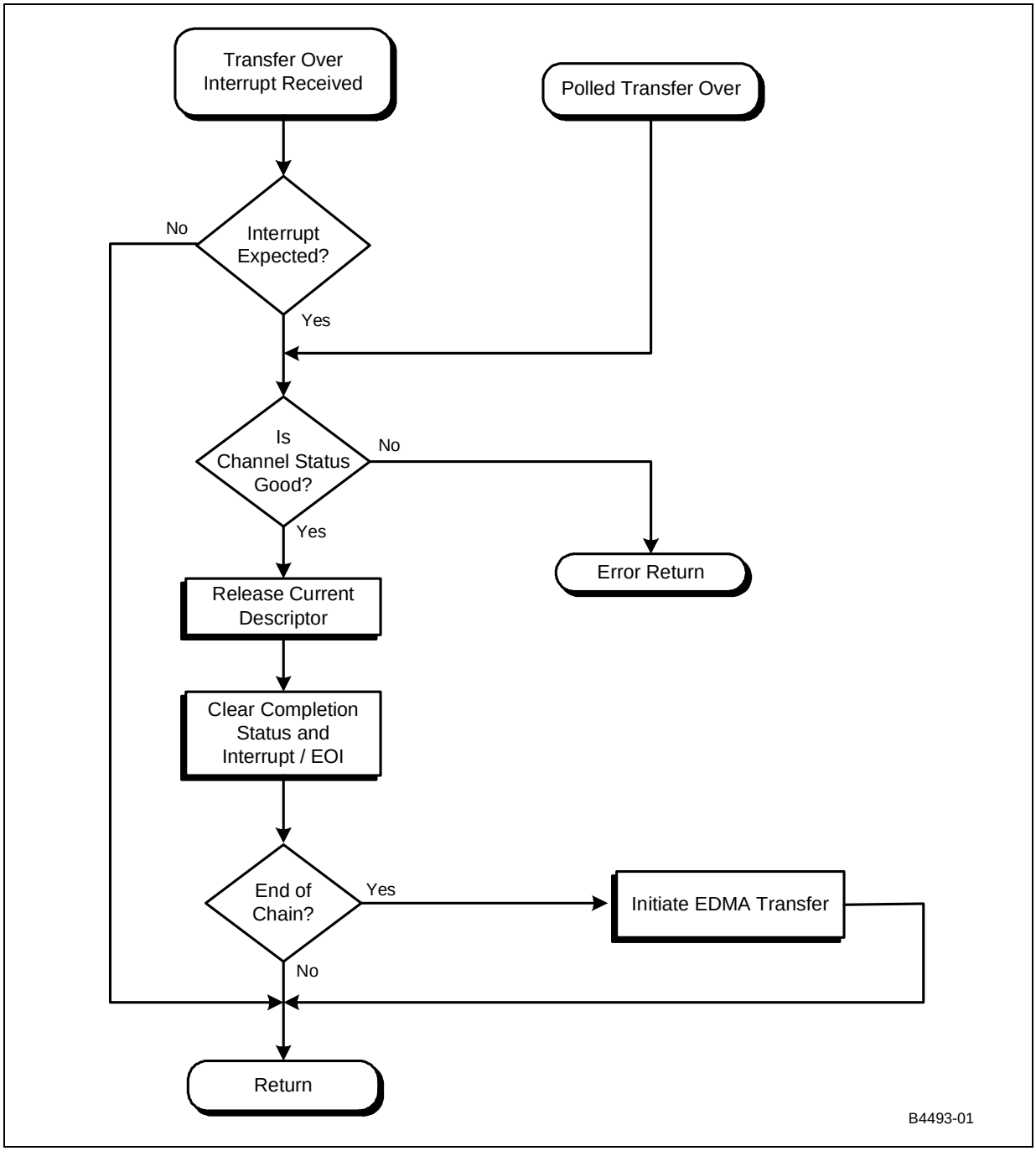


Figure 35. Completion Flow Chart



4.0 Power Requirements and Interface Signals

This chapter covers areas related to the IMCH's power and signal interfaces:

- Description of the power supply requirements
- Power wells and the platform expectations for connecting the power wells
- Power planes and the associated controlled and controlling logic
- Interface signal descriptions
- Interface signal power well and state information during reset and the first clock after reset
- Reset straps and function

Table 59. Terminology

Term	Definition
Power Supply	The external voltages required to operate a component.
Power Plane	A group of like voltages connected to a particular power supply.
Power Well	Power planes which operate in normal and power management states.

Table 60. Summary of Interfaces (Sheet 1 of 2)

Signal Group	Description	Section	Maximum Total Signals[1]
Power and Ground	Power and Ground Planes	Section 4.3	601
Processor Interface	Front Side Bus (FSB) and Processor Control Signals	Section 4.4.1.10	162
Memory Bus Interface	Memory Bus Interfaces and Control	Section 4.4.1.11	155
PCI Express* Interface	One x8 and one x4 PCI Express Interfaces	Section 4.4.1.12	56
SMBus Interface	System Management and SMBus Interfaces	Section 4.4.1.13	8
Miscellaneous	Clocks, Resets, and Miscellaneous Signals	Section 4.4.1.14	17
LPC Interface	Low Pin Count Bus Interface Signals	Section 4.4.1.15	8
USB Interface	Port Universal Serial Bus Interface Signals	Section 4.4.1.16	14
Serial ATA Interface	Serial ATA port interface Signals	Section 4.4.1.17	35
UART Interface	Universal Async Receive and Transmit Interface	Section 4.4.1.18	17
Interrupt Interface	Interrupts Plus one Serial Interrupt	Section 4.4.1.19	9
Power Management	All Power Management Features	Section 4.4.1.20	15
RTC and WDT	Crystal Input 1 and 2 and Watchdog Timer	Section 4.4.1.21	3
PCI Interface	32-bit, 33 MHz PCI Bus	Section 4.4.1.22	50

Table 60. Summary of Interfaces (Sheet 2 of 2)

Signal Group	Description	Section	Maximum Total Signals[1]
General Purpose I/O	General Purpose I/O signals	Section 4.4.1.23	38
Debug Interface	TAP and XDP Interface	Section 4.4.1.24	14

Total signals pins = 650[1,2]

1. Maximum Total Signals count includes multiplexed signals. Total signal pins is not equal to the summation of the maximum signal counts.
2. Power and ground planes are not included in total signal pin count.

4.1 Power Supply Requirements

Table 61 provides a summary of the required supply levels and tolerances. See the *External Design Specification Addendum* for full details.

Table 61. Power Supply Requirements

Nominal Voltage	Voltage Tolerance	Interface Supplied	Comments
1.05 V	+/- 3%	FSB	Vtt on the processor interface
1.5 V	+/- 3%	Core, PCI Express, SATA, USB Core and Sus	Vcc for core logic, PCI Express and SATA interfaces, and USB Core and USB SUS plane
1.8 V	+/- 5%	DDR2	Vcc on DDR2 interface and DIMM slots.
2.5 V	+/- 5%	Core	Vcc for core internal logic
3.3 V	+/- 5%	SMB, I/Os, GPIOs, MISC, RTC	Utilized for ESD protection on 3.3 V tolerant interfaces: SMB, RSTIN#, PE_HPINTR#, GPIOs,Power Management and Reset Circuitry, and Real Time Clock
5 V	+/- 5%	PCI, USB Ref	Reference for 5 V tolerance on PCI and USB signals
2.0-3.0 V	NA	Battery	RTC battery voltage

4.2 Power Wells

This section provides an overview of the power wells and the platform expectations for connecting the power wells. There are shown in Table 62.

Table 62. Power Wells

Well	Usage
Core	This well is enabled during full power configuration.
Suspend/ Resume	This well is enabled during full power configuration and in certain sleep states. This power is not expected to be shut off unless the system is unplugged.
RTC	This power is not expected to be shut off unless the RTC battery is removed or drained. When main power is not available, an external RTC battery, typically a lithium-type coin cell, will be used.

Note: Refer to the Power Management Chapter for additional details on sleep and power states.

Figure 36. Intel® Pentium® M Processor Based Platform Power Delivery

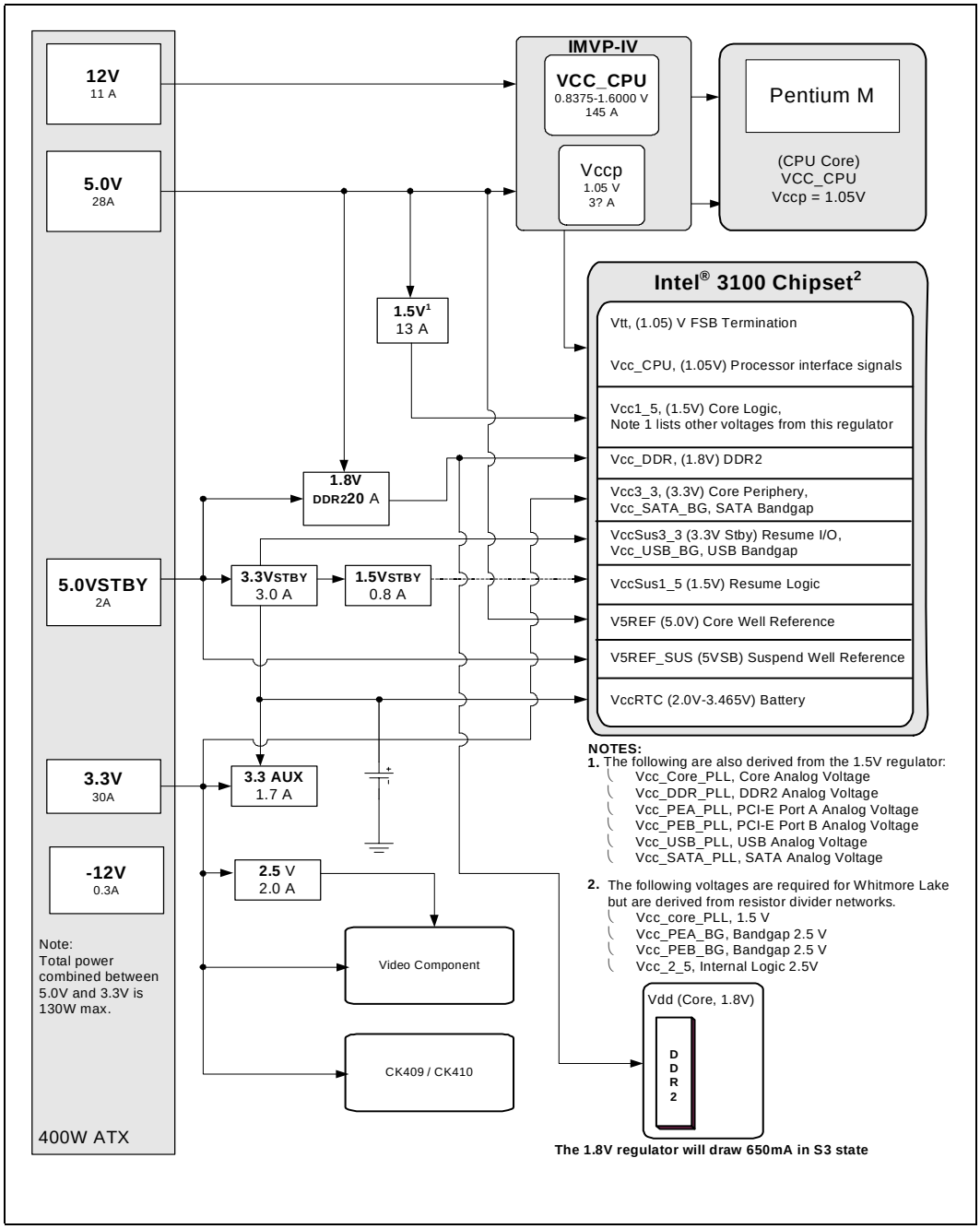
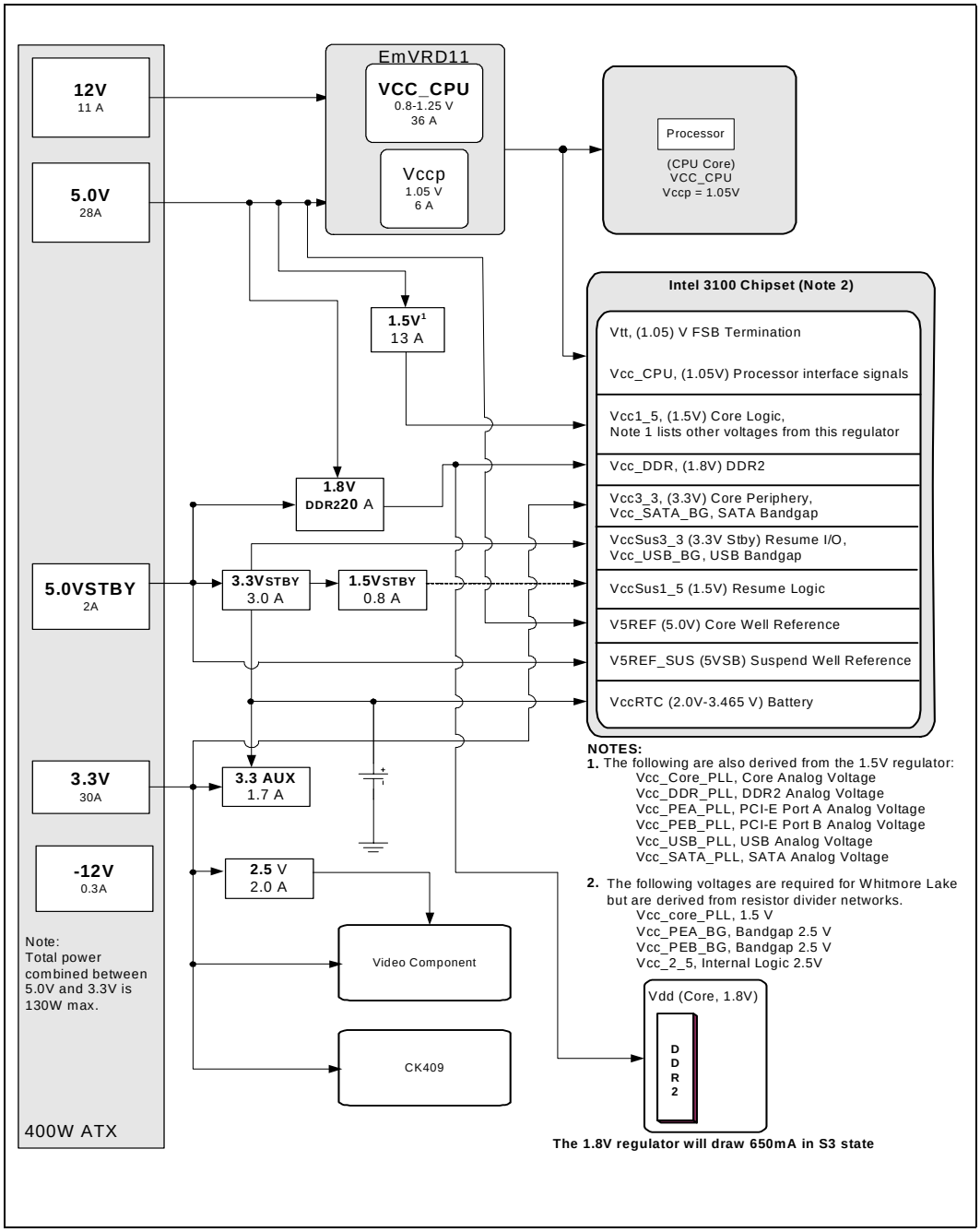


Figure 37. Dual-Core Intel® Xeon® Processor LV Based Platform Power Delivery



4.3 Power Planes

The following table summarizes the power and ground supplies.

Table 63. Power and Ground Planes (Sheet 1 of 2)

Signal Name	I/O	Well	Pkg Balls	Description
VTT	I	Core	38	Power: Power supply pins for the front side bus (FSB) interface. Voltage is the same as processor VTT.
VCC1_5	I/O	Core	112	Power: 1.5 V supply pins for the core well.
VCCSUS1_5	I/O	Suspend/ Resume	3	Power: 1.5 V power pins for the resume well logic. This power is not expected to be shut off unless the system is unplugged. This voltage may be generated internally by a regulator. If generated internally, these pins should not be connected to an external supply.
VCC_CORE_PLL	I/O	Core	1	1.5 V analog VCC supply.
VCC_DDR_PLL	I/O	Core	1	1.5 V analog VCC supply.
VCC_PEA_PLL	I/O	Core	1	1.5 V analog VCC supply.
VCC_PEA_BG	I/O	Core	1	2.5 V PCI Express Band-Gap Port A power
VCC_PEB_PLL	I/O	Core	1	1.5 V Power for PCI Express Port B PLL
VCC_PEB_BG	I/O	Core	1	3.3 V PCI Express Band-Gap Port B Power
VCC_SATA_PLL	I/O	Core	1	1.5 V Analog power supply. This signal is used for the SATA PLL. This voltage is required even if SATA is not used.
VCC_SATA_BG	I/O	Core	1	Analog 3.3 V Power for SATA AFE Reference Circuitry
VCC_USB_PLL	I/O	Core	1	1.5 V analog power supply. This signal is used for the USB PLL. This voltage is required even if USB is not used.
VCC_USB_BG	I/O	Suspend/ Resume	1	Analog 3.3 V Power for USB AFE Reference Circuitry
VCC_DDR	I/O	Core	46	Power: 1.8 V power supply pins for the DDR2 interface.
VCC2_5	I/O	Core	2	2.5 V power supply pins
VCC3_3	I/O	Core	29	3.3 V signal supply pins
VCCSUS3_3	I/O	Suspend/ Resume	14	3.3 V power pins for resume planes
VCCRTC	I/O	RTC	1	3.3 V power supply pin for the RTC Well. This power is not expected to be shut off unless the RTC battery is removed or drained. ***Note:*** Implementations must not attempt to clear CMOS by using a jumper to pull VccRTC low. Clearing CMOS can be done by using a jumper on RTEST# or GPI.
VCC_CPU	I/O	Core	3	Powered by the same supply as the processor I/O voltage, 1.05 V. This supply is used to drive the processor interface signals.
V5REF	I/O	Core	2	5 V reference
V5REFSUS	I/O	Suspend/ Resume	1	Reference for 5 V tolerance on resume well inputs
VSS	I/O	Core	334	Ground: Digital ground pins.
VSS_CORE_PLL	I/O	Core	1	Ground: Analog ground
VSS_PEA_PLL	I/O	Core	1	Ground: Analog ground.
VSS_PEA_BG	I/O	Core	1	PCI Express Band-Gap Port A analog Ground
VSS_PEB_BG	I/O	Core	1	PCI Express Band-Gap Port B analog Ground

Table 63. Power and Ground Planes (Sheet 2 of 2)

Signal Name	I/O	Well	Pkg Balls	Description
VSS_SATA_BG	I/O	Core	1	Analog Ground for SATA AFE (Analog Front End) reference circuitry
VSS_USB_BG	I/O	Suspend/ Resume	1	Analog Ground for 3.3 V USB AFE reference circuitry, band gap
Total Power and Ground Signals = 601				

4.4 Signal Information

- This section lists all of the signals that connect to interface balls on the chip package. The I/O field indicates the functional mode for the pin during normal operation.
- Cell I/O Class field indicates which class of I/O cell is used for the pin.
- Package Balls field indicates the number of balls on the package devoted to the listed signal group.

There are a total of 650 signal pins. Many signal pins are dual function resulting in the summation of total number of potential signals for all interfaces exceeding the actual total signal pin count. The power and ground pins are not included in the total signal pin count.

4.4.1 Signaling Technology

This section provides an overview of the various signaling technology utilized, with focus on differing supply voltages and termination requirements. Refer to the *External Design Specification Addendum* for further details on signaling technologies and interface pin specifications.

4.4.1.1 AGTL+, AGTL+2x, AGTL+4x Signals

The AGTL+ technology is a generational advancement on the Gunning Transceiver Logic design, and was introduced by Intel with the launch of the Pentium® III processor family. The suffix after the AGTL+ refers to the data rate of that pin:

- AGTL+2X - Double-pump clocking. Addressing at 2x of HCLK
- AGTL+4X - Quad-pump clocking. Data transfers at 4x of HCLK

These signals operate off of a 1.05 V supply, utilize a reference voltage at 2/3 of the rail, and are active low. Transceivers behave as modified open-drains when driving; in that they actively drive the low (logically asserted) voltage, but only “kick” the high (logically deasserted) voltage prior to releasing the bus to be held by on-board termination resistors. AGTL+ technology requires termination at “both ends” of the bus.

The AGTL+ bus termination resistors are often included within the processors and chipsets, as is the case with the Intel 3100 chipset, and all the processors which are paired to it.

4.4.1.2 Host Clock Signal Level (HCSL) Signals

The processor and this chip’s reference clocks are HCSL, low-voltage differential CMOS technology. These signals have a low voltage swing (0.7 V), with tightly matched rise and fall times, and a 150 ps jitter requirement.

4.4.1.3 CMOS and SCHMITT Signals

The CMOS interface signals are handled by generic CMOS transceiver designs, operating on a full swing from rail to rail, with nominal threshold centered between them. The Intel® 3100 Chipset implements different CMOS voltage types. Table 64 lists the CMOS types and their associated voltages.

Some signals are designated "SCHMITT" in the signal description lists, indicative of a standard CMOS signaling technology, but designed to compensate for slow edges with an internal SCHMITT trigger circuit. The IMCH implements different SCHMITT voltage types. Table 64 lists the SCHMITT types and their associated voltages.

Table 64. CMOS and SCHMITT Signals Types

Type Name	Signal Technology	Voltage
CMOS1_05	CMOS	1.05
CMOS1_5	CMOS	1.5
CMOS1_8	CMOS	1.8
CMOS3_3	CMOS	3.3
CMOS5_0	CMOS	5.0
SCHMITT1_5	SCHMITT	1.5
SCHMITT3_3	SCHMITT	3.3
SCHMITT5_0	SCHMITT	5.0

4.4.1.4 SSTL-2 Signals

The SSTL-2 technology is used for the source-synchronous DDR2 interface signals. These signals operate on the same 1.8 Volt supply as the DDR2 devices themselves, and are similar to standard CMOS pins.

4.4.1.5 Open Drain (OD) Signals

The device can drive a logic-0 on the pin, but relies on an external pull-up resistor to attain a logic-1 level. The voltage associated with logic-1 is specified for each OD type signal, and the design relies on the platform architect to ensure that these pins are not inadvertently pulled to an unsupported voltage level. For bidirectional signals (I/OD), the external device will signal logic-0 and logic-1 as described.

4.4.1.6 PCI Express* (PCIE) Signals

The PCI Express interface signals are driven by transceivers designed specifically for high-speed serial communication. All PCI Express signals are fully differential, and actually operate in a current mode, rather than a voltage mode. These interfaces support A/C coupling to facilitate communication across independent power supply domains, and signal at a rate well above the flight time of the interface.

4.4.1.7 SATA Signals

The SATA interface is a high speed serial link, which supports 1.5 Gbits/s. This technology consists of a receive and transmit.

4.4.1.8 USB Signals

The USB interface supports both USB1.1 and USB 2.0 signaling. USB1.1 has low-speed mode (1.5 Mbps) and full-speed mode (12 Mbps). USB 2.0 works at high speed LVDS mode with a data rate of 480 Mbps.

4.4.1.9 Analog Signals

The callout of analog references a signal type which does not follow standard digital logic levels. These signals include impedance compensation, power, ground, and other special characteristic signals.

4.4.1.10 Processor Interface

Table 65. Processor Bus Interface (Sheet 1 of 3)

Signal Name	I/O	Class	Pkg Balls	Description
ADS#	I/O	AGTL+	1	**Processor Address Strobe:** Indicates valid address on the HA pins. The system bus owner asserts ADS# to indicate the first of two cycles of a request phase.
AP[1:0]#	I/O	AGTL+	2	**Processor Address Parity:** Provides parity protection for the HA and HREQ signals. The AP[1:0]# lines are driven by the request initiator and provide parity protection for the Request Phase signals. AP[1:0]# are common clock signals and are driven one common clock after the Request Phase. Address parity is correct if there are an even number of electrically low signals (low voltage) in the set consisting of the covered signals plus the parity signal. ***Note:*** Only connects to HA[35:3]#.
BINIT#	I	AGTL+	1	**Processor Bus Initialization:** This latched signal indicates an unrecoverable error and can be driven by the processor.
BNR#	I/O	AGTL+	1	**Processor Block Next Request:** Blocks new requests by the current bus owner. This signal is used to dynamically control the system bus pipeline depth.
BPRI#	O	AGTL+	1	**Processor Priority Agent Bus Request:** Used to arbitrate for ownership of the processor bus. The IMCH is the only Priority Agent on the system bus. It asserts this signal to obtain the ownership of the address bus. The IMCH has priority over symmetric bus requests and will cause the current symmetric owner to stop issuing new transactions unless the HLOCK# signal is asserted.
BREQ0#	I/O	AGTL+	1	**Processor Bus Request 0:** The IMCH pulls the processor bus BREQ0# signal low during CPURST#. The signal is sampled by the processors on the active-to-inactive transition of CPURST#. The minimum setup time for this signal is 4 HCLKs. The minimum hold time is 2 HCLKs and the maximum hold time is 20 HCLKs. BREQ[0]# should be tristate after the hold time requirement has been satisfied.
BREQ1#	I	AGTL+	1	**Processor Bus Request 1:** The IMCH does not drive this pin. It is used by the processors for symmetric bus arbitration. The IMCH samples this signal during run time only.
CPURST#	O	AGTL+	1	**Processor Bus Reset:** The IMCH asserts CPURST# while RSTIN# is asserted and for approximately 1 ms after RSTIN# is deasserted. The CPURST# allows the processors to begin execution in a known state.
DBSY#	I/O	AGTL+	1	**Processor Data Bus Busy:** Used by the data bus owner to hold the data bus for transfers requiring more than one cycle.

Table 65. Processor Bus Interface (Sheet 2 of 3)

Signal Name	I/O	Class	Pkg Balls	Description
DEFER#	O	AGTL+	1	**Processor Bus Defer:** Signals that the IMCH will terminate the transaction currently being snooped with either a deferred response or with a retry response.
DP[3:0]#	I/O	AGTL+	4	**Processor Data Bus Parity:** The DP[3:0]# signals provide parity protection for HD[63:0]#. The DP[3:0]# signals are common clock signals and are driven one common clock after the data phases they cover. DP[3:0]# are driven by the same agent driving HD[63:0]#. Data parity is correct if there are an even number of electrically low signals (low voltage) in the set consisting of the covered signals plus the parity signal.
DINV[3:0]#	I/O	AGTL+4x	4	**Processor Dynamic Bus Inversion:** Driven along with the HD[63:0]# signals. Indicates when the associated signals are inverted. DINV[3:0]# are asserted such that the number of data bits driven electrically low (low voltage) within the corresponding 16-bit group never exceeds eight.
DRDY#	I/O	AGTL+	1	**Processor Data Ready:** Asserted for each cycle that data is transferred.
HA[35:3]#	I/O	AGTL+2x	33	**Processor Address Bus:** HA[35:3]# connect to the system address bus. During processor cycles, HA[35:3]# are inputs. The IMCH drives HA[35:3]# during snoop cycles.
HADSTB[1:0]#	I/O	AGTL+2x	2	**Processor Address Bus Strobe:** The source synchronous strobes are used to transfer HA[35:3]# and HREQ[4:0]# at the 2x transfer rate.
HACVREF	I	Analog	1	**Processor Address/Control Bus Reference Voltage:** Reference voltage input for the Address and Control signals of the Host AGTL+ interface.
HCLKINn, HCLKINp	I	HCSL	2	**Differential Processor Clock Input(s):** These pins receive a differential host clock from the external clock synthesizer. This clock is used by all the IMCH logic in the host clock domain.
HD[63:0]#	I/O	AGTL+4x	64	**Processor Data Bus:** These signals are connected to the system data bus.
HDSTBn[3:0]#, HDSTBp[3:0]#	I/O	AGTL+4x	8	**Processor Data Bus Strobe:** The differential source synchronous strobes are used to transfer HD[63:0]# and DINV[3:0]# at the 4x transfer rate. Strobe Data Bits: HDSTBp[3]#, HDSTBn[3]#, HD[63:48]#, DINV[3]# HDSTBp[2]#, HDSTBn[2]#, HD[47:32]#, DINV[2]# HDSTBp[1]#, HDSTBn[1]#, HD[31:16]#, DINV[1]# HDSTBp[0]#, HDSTBn[0]#, HD[15:0]#, DINV[0]#
HDVREF[1:0]	I	Analog	2	**Processor Bus Data Reference Voltage:** Reference voltage input for the 4x Data Signals of the Host AGTL+ interface.
HIT#	I/O	AGTL+	1	**Processor Bus Hit:** Indicates that a caching agent holds an unmodified version of the requested line. Also, driven in conjunction with HITM# by the target to extend the snoop window.
HITM#	I/O	AGTL+	1	**Processor Bus Hit Modified.** Indicates that a caching agent holds a modified version of the requested line and that this agent assumes responsibility for providing the line. HITM# is driven in conjunction with HIT# to extend the snoop window.
HLOCK#	I	AGTL+	1	**Processor Lock:** Indicates a transaction must occur autonomously. All system bus cycles that are sampled with the assertion of HLOCK# and ADS#, until the negation of HLOCK# is atomic.

Table 65. Processor Bus Interface (Sheet 3 of 3)

Signal Name	I/O	Class	Pkg Balls	Description
HREQ[4:0]#	I/O	AGTL+2x	5	**Processor Request Command:** Driven by current bus owner to define active transaction type. HREQ[4:0]# are transferred at the 2x rate. Asserted by the requesting agent during both halves of a Request Phase. In the first half, the signals define the transaction type to a level of detail that is sufficient to begin a snoop request. In the second half, the signals carry additional information to define the complete transaction type.
HTRDY#	O	AGTL+	1	**Processor Bus Target Ready:** Indicates that the target of the processor transaction is able to enter the data transfer phase.
HCRES0	I/O	Analog	1	**Processor Compensation Resistor Return:** Common VSS return for the processor bus compensation resistors on HODTCRES and HSLWCRES.
HSLWCRES	I	Analog	1	**Compensation Resistor:** Compensation for processor bus slew rate.
HODTCRES	I	Analog	1	**Compensation Resistor:** Compensation for processor bus on-die termination
MCERR#	I/O	AGTL+	1	**Machine Check Error:** Connected to processor MCERR#. May be asserted by the IMCH or the processors to indicate an unrecoverable error without a bus protocol violation. The following assertion options are configurable at a system level. • Enable/Disable • Asserted, if configured, along with external processor IERR# for processor internal errors • Asserted, if configured, by the request initiator of a bus transaction after it observes and error • Asserted by any bus agent when it observes an error in a bus transaction
RS[2:0]#	O	AGTL+	3	**Processor Bus Response Status:** indicate the type of response according to the following table: **RS[2:0] Response Type:** 000 Idle state 001 Retry response 010 Deferred response 011 Reserved (invalid) 100 Hard Failure (never generated) 101 No data response 110 Implicit Writeback 111 Normal data response
RSP#	O	AGTL+	1	**Processor Bus Response Parity:** Provides parity protection for RS[2:0] signals. RSP# is always driven by the IMCH, and is valid on all clocks. Parity is correct if there are an even number of electrically low signals (low voltage) in the set consisting of the covered signals plus the parity signal.
Total signals in group = 149				

Table 66. Processor Interface (Sheet 1 of 2)

Signal Name	I/O	Class	Pkg Balls	Description
A20M#	O	CMOS1_05	1	**Mask A20:** A20M# will go active based on either setting the appropriate bit in the Port 92h register or based on the A20GATE input being active. A20M# will go inactive due to an INIT#.
CPUSLP#	O	CMOS1_05	1	**CPU Sleep:** This signal puts the processor into a state that saves substantial power vs. the Stop-Grant state. However, during that time, no snoops occur.
FERR#	I	CMOS1_05	1	**Numeric Coprocessor Error:** This signal is tied to the coprocessor error signal on the processor. FERR# is only used if the coprocessor error reporting function is enabled in the General Control Register (D31:F0:Offset D0.bit 5). If FERR# is asserted, IRQ13 is sent to the interrupt controller unit. It is also used to gate the IGNNE# signal to ensure that IGNNE# is not asserted to the processor unless FERR# is active. FERR# requires an external weak pull-up to ensure a high level when the coprocessor error function is disabled. ***Note:*** FERR# can optionally be used in some states for notification by the processor of pending interrupt events.
IGNNE#	O	CMOS1_05	1	**Ignore Numeric Error:** This signal is connected to the ignore error pin on the processor. IGNNE# is only used if the coprocessor error reporting function is enabled in the General Control Register (D31:F0:Offset D0.bit 5). If FERR# is active, indicating a coprocessor error, a write to the Coprocessor Error Register (F0h) causes the IGNNE# to be asserted. IGNNE# remains asserted until FERR# is negated. If FERR# is not asserted when the Coprocessor Error Register is written, the IGNNE# signal is not asserted.
INIT#	O	CMOS1_05	1	**Initialization:** INIT# is asserted for 16 PCICLK clocks to reset the processor. The chip can be configured to support processor Built In Self Test (BIST).
INIT3_3V#	O	CMOS3_3	1	**Initialization 3.3 V:** This is the identical 3.3 V copy of INIT#.
INTR	O	CMOS1_05	1	**Processor Interrupt:** INTR is asserted to signal the processor that an interrupt request is pending and needs to be serviced. It is an asynchronous output normally driven low.
NMI	O	CMOS1_05	1	**Non-Maskable Interrupt:** NMI is used to force a non-maskable interrupt to the processor, if configured, when either SERR# or ISA IOCHK# is asserted. The processor detects an NMI as a rising edge on NMI. NMI is reset by setting the corresponding NMI source enable/disable bit in the NMI Status and Control Register.
SMI#	O	CMOS1_05	1	**System Management Interrupt:** SMI# is an active low output synchronous to PCICLK that is asserted in response to one of many enabled hardware or software events.
STPCLK#	O	CMOS1_05	1	**Stop Clock Request:** STPCLK# is an active-low output synchronous to PCICLK that is asserted in response to one of many hardware or software events. When the processor samples STPCLK# asserted, it responds by stopping its internal clock.
RCIN#	I	CMOS3_3	1	**Keyboard Controller Reset Processor:** The keyboard controller can generate INIT# to the processor. This saves the external OR gate of other sources of INIT#. When the chip detects the assertion of this signal, INIT# is generated for 16 PCICLK clocks. ***Note:*** Will ignore RCIN# assertion during transitions to the S3 state.

Table 66. Processor Interface (Sheet 2 of 2)

Signal Name	I/O	Class	Pkg Balls	Description
A20GATE	I	CMOS3_3	1	**A20 Gate:** A signal from the keyboard controller. Acts as an alternative method to force the A20M# signal active. Saves the external OR gate needed with various other chipsets.
CPUPWRGD/GPO49	O	OD	1	**CPU Power Good:** This signal must be connected to the processor's PWRGOOD input in order to allow for Intel SpeedStep® technology support. This signal is kept high during an Intel SpeedStep technology state transition to prevent loss of processor context. This is an open-drain output signal (external pull-up resistor required) that represents a logical AND of PWROK and VRMPWRGD signals. When an open drain resistor is hooked up to the CPU/IO power plane, the CPU/IO voltage shall not exceed the Core reference voltage (1.5 V). This signal may optionally be configured as a GPO[49].
Total signals in group = 13				

4.4.1.11 Memory Interface DDR2

Table 67. DDR2 Memory Bus Interface

Signal Name	I/O	Class	Pkg Balls	Description
DDR_BA[2:0]	O	CMOS1_8	3	**DDR2 Channel Bank Address:** The DDR bank address signals. These signals are outputs of the IMCH and select which bank within a row is selected.
DDR_CAS#	O	CMOS1_8	1	**DDR2 Channel Column Address Strobe:** Used to indicate a valid column address and initiate a transaction.
DDR_CMDCLKp[3:0], DDR_CMDCLKn[3:0]	O	CMOS1_8	8	**DDR2 Channel Command Clock (Differential):** The DDR command clocks used by the DDR2 DRAMs to latch MA[14:0], BA[2:0], RAS#, CAS#, WE#, CKE#, and CS# signals.
DDR_CS[7:0]#	O	CMOS1_8	8	**DDR2 Channel Chip Select and ODT:** Used to indicate to which DRAM device cycles are targeted.
DDR_MA[14:0]	O	CMOS1_8	15	**DDR2 Channel Memory Address:** The DDR2 memory address signals.
DDR_RAS#	O	CMOS1_8	1	**DDR2 Channel Row Address Strobe:** Used to indicate a valid row address and open a row.
DDR_WE#	O	CMOS1_8	1	**DDR2 Channel Write Enable:** Used to indicate a write cycle.
DDR_DQ[63:0]	I/O	SSTL-2	64	**DDR2 Channel Data Bus:** The DDR2 data bus provides the data to/from the DRAM devices.
DDR_CB[7:0]	I/O	SSTL-2	8	**DDR2 Channel Check Bits:** These check bits are required to provide ECC support.
DDR_DQSp[17:0], DDR_DQSn[17:0]	I/O	SSTL-2	36	**DDR2 Channel Data Strobes (Differential):** Each data strobe is used to strobe a set of four or eight data signals (depending on whether x4 or x8 DRAM devices are used).
DDR_VREF	I	Analog	1	**DDR2 Channel Voltage Reference:** DDR2 Reference voltage input.
DDR_CKE[3:0]	O	CMOS1_8	4	**DDR2 Clock Enable:** Independent per DIMM slot
DDR_CRES0	I/O	Analog	1	**DDR2 Compensation Resistor Return:** Common return for DDR interface compensation resistors on DDRSLWCRES and DDRIMPCRES.
DDR_SLWCRES	I/O	Analog	1	**Compensation Resistor:** Slew rate compensation for DDR2 interface
DDR_IMPCRES	I/O	Analog	1	**Compensation Resistor:** Impedance compensation for DDR2 interface

Table 67. DDR2 Memory Bus Interface

Signal Name	I/O	Class	Pkg Balls	Description
DDR_RES1	I/O	Analog	1	**DDR2 Resistor**: Additional compensation resistor (resistor 1) for DDR2 interface
DDR_RES2	I/O	Analog	1	**DDR2 Resistor**: Additional compensation resistor (resistor 2) for DDR2 interface
Total signals in group = 155				

4.4.1.12 PCI Express* Interfaces

PCI Express ports are configurable to accommodate different applications. Figure 38 demonstrates how the IMCH's 1 x8 port can be trained to two x4 or two x1 ports. Figure 39 demonstrates how the IICH's 1 x4 port can be trained to four x1 ports.

Figure 38. IMCH's 1 x8 PCI Express* PEA Link Configuration Examples

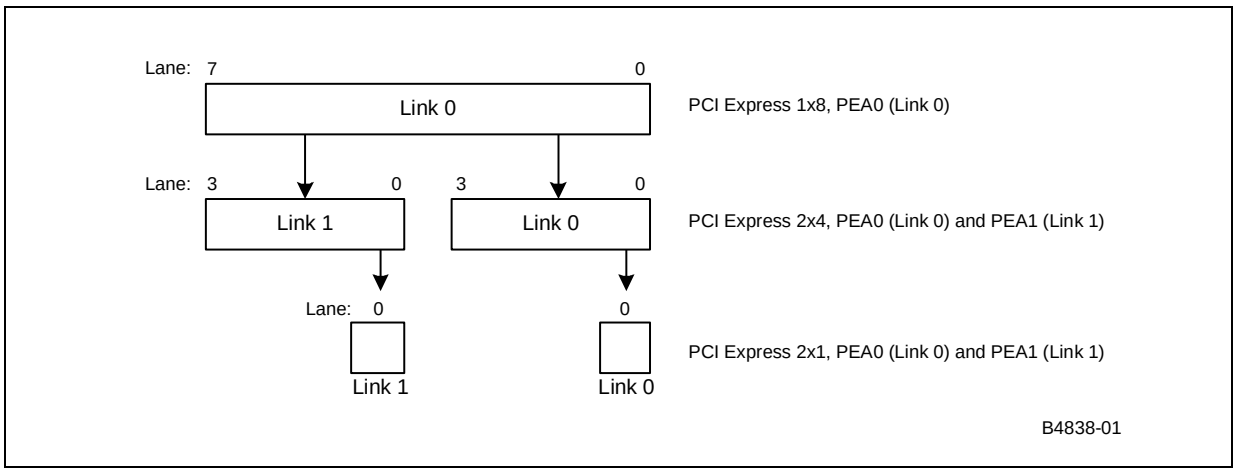
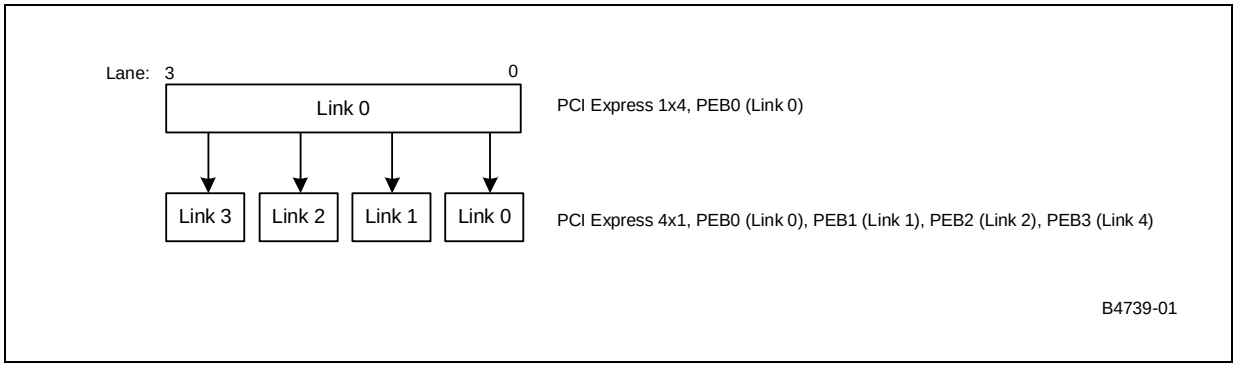


Figure 39. IICH's 1 x4 PCI Express* Link Configuration Examples



4.4.1.12.1 PCI Express* Naming Convention Definitions

PCI Express signal names in Table 68 and Table 69 follows the PCI Express signal naming convention:

PE<port><link #>_<T/R><p/n>[bit]

Where:

- PE: Defines PCI Express
- <port>:
  - — Port “A” supported configurations: 1x8 or 2x4 or 2x1
  - — Port “B” supported configurations: 1x4 or 4x1
- <Link #>: Defines the link in use
- <T/R>: Defines a transmit (T) or receive (R) signal
- <p/n>: Defines the polarity of the differential signal “p(+), n(–)”
- [bit]: Defines the bits used in the link

Table 68. PCI Express* Interface (Sheet 1 of 2)

Signal Name	I/O	Class	Pkg Balls	Description
PEA0_Tp[7:0], PEA0_Tn[7:0]	O	PCIE	16	**PCI Express Interface Port A Output (transmit) Data Pair** (differential). These signals are output (TX) from the IMCH’s perspective, and must be connected to the input (receive or RX) signals of the other PCI Express device. ***Note:*** 1. The signals for this 1x8 interface can be trained to 2x4 or 2x1 ports. 2. y = n or p **1 x8 Interface Configuration:** • PEA0_Ty[7:0] = Link 0 **2 x4 Interface Configuration:** • PEA1_Ty[7:4] = Link 1 • PEA0_Ty[3:0] = Link 0 **2 x1 Interface Configuration:** • PEA1_Ty[4] = Link 1 • PEA0_Ty[0] = Link 0
PEA0_Rp[7:0], PEA0_Rn[7:0]	I	PCIE	16	**PCI Express Interface Port A Input (receive) Data Pair** (differential). These signals are input (RX) from the IMCH’s perspective, and must be connected to the output (transmit or TX) signals of the other PCI Express device. ***Note:*** 1. The signals for this 1x8 interface can be trained to 2x4 or 2x1 ports. 2. y = n or p **1 x8 Interface Configuration:** • PEA0_Ry[7:0] = Link 0 **2 x4 Interface Configuration:** • PEA1_Ry[7:4] = Link 1, • PEA0_Ry[3:0] = Link 0 **2 x1 Interface Configuration:** • PEA1_Ry[4] = Link 1, • PEA0_Ry[0] = Link 0
PEA_CLKp PEA_CLKn	I	HCSL	2	**PCI Express Port A Clock:** clock reference input (differential)
PEA_RCOMPO	O	Analog	1	**PCI Express A Port Compensation (out):** Used to calibrate the PCI Express high speed serial input/output buffers
PEA_ICOMPI	I	Analog	1	**PCI Express A Port Compensation:** Used to calibrate the PCI Express high speed serial input/output buffers

Table 68. PCI Express* Interface (Sheet 2 of 2)

Signal Name	I/O	Class	Pkg Balls	Description
PEB0_Tp[3:0], PEB0_Tn[3:0]	O	PCIE	8	**PCI Express Interface Port B Output (transmit) Data Pair** (differential). These signals are output (TX) from the IICH's perspective, and must be connected to the input (receive or RX) signals of the other PCI Express device. ***Note:*** 1. The signals for this 1x4 interface can be trained to 4x1 or 1x4 ports. 2. y = n or p **1 x4 Interface Configuration:** • PEB0_Ty[3:0]y = Link 0, **4 x1 Interface Configuration:** • PEB3_Ty[3] = Link 3, • PEB2_Ty[2] = Link 2, • PEB1_Ty[1] = Link 1, • PEB0_Ty[0] = Link 0
PEB0_Rp[3:0], PEB0_Rn[3:0]	I	PCIE	8	**PCI Express Interface Port B Input (receive) Data Pair** (differential). These signals are input (RX) from the IICH's perspective, and must be connected to the output (transmit or TX) signals of the other PCI Express device. ***Note:*** 1. The signals for this 1x4 interface can be trained to 4x1 or 1x4 ports. 2. y = n or p **1 x4 Interface Configuration:** • PEB0_Ry[3:0] = Link 0, **4 x1 Interface Configuration:** • PEB3_Ry[3] = Link 3, • PEB2_Ry[2] = Link 2 • PEB1_Ry[1] = Link 1 • PEB0_Ry[0] = Link 0
PEB_CLKp, PEB_CLKn	I	HCSL	2	**PCI Express Port B Clock:** clock reference input (differential)
PEB_RCOMP	O	Analog	1	**PCI Express Port B Compensation (out):** Used to calibrate the PCI Express high speed serial input/output buffers
PEB_ICOMPI	I	Analog	1	**PCI Express Port B Compensation:** Used to calibrate the PCI Express high speed serial input/output buffers
Total signals in PCI Express = 56				

4.4.1.13 System Management and SMBus Interfaces

See Chapter 26.0, "Device 31, Function 3: SMBus Controller Functional Description," for full behavioral descriptions of the IMCH's SMBus.

Table 69. SMBus Interface

Signal Name	I/O	Class	Pkg Balls	Description
SMBDATA	I/O	OD	1	IICH **SM Bus Data:** External pull-up required.
SMBCLK	I/O	OD	1	IICH **SM Bus Clock:** External pull-up required.
SMBALERT#/ GPI11	I	CMOS3_3	1	**SMBus Alert:** This signal is used to wake the system or generate SMI#. ***Note:*** If this bit is not needed as SMBALERT#, it can be used as a GPI[11].
INTRUDER#	I	CMOS3_3	1	**Intruder Detect:** Detects if the system case has been opened. Can be set to disables the system if the box is detected open. This input signal is in the RTC well. This signal's status is readable, so it can be used like a GPI if the Intruder switch is not needed.
SMLINK[1:0]	I/O	OD	2	**System Management Link:** SMBus link to optional external system management ASIC or LAN controller. External pull-ups are required. Note that SMLINK0 corresponds to an SMBus Clock signal, and SMLINK1 corresponds to an SMBus Data signal.
SMBSDA	I/O	OD	1	IMCH **SMBus Data:** Data signal for the SMBus interface (3.3 V) Internal PU, 8 k to 80 k, typically 15 k ohms.
SMBSCL	I/O	OD	1	IMCH **SMBus Clock:** Clock signal for the SMBus interface (3.3 V) Internal PU, 8 k to 80 k, typically 15 k ohms.
Total signals in group = 8				

4.4.1.14 Clocks, Resets, and Miscellaneous Signals

Table 70. Clocks, Resets, and Miscellaneous Signals (Sheet 1 of 2)

Signal Name	I/O	Class	Pkg Balls	Description
CPU_SEL[2:0]	I	SCHMITT1_5	3	**Processor Interface Select:** The processor select encodings are used at power-on to select the frequency mode of the PLL circuitry. Unreliable behavior will result from an attempt to set a Reserved setting. CPU_SEL[2:0] = 000 =Reserved CPU_SEL[2:0] = 001 = 133 MHz FSB, DDR2/FSB Ratio 3:2 CPU_SEL[2:0] = 010 = Reserved CPU_SEL[2:0] = 011 = 167 MHz FSB, DDR2/FSB Ratio 6:5 CPU_SEL[2:0] = 100 = Reserved CPU_SEL[2:0] = 101 = 100 MHz FSB, DDR2/FSB Ratio 2:1 CPU_SEL[2:0] = 110 = Reserved CPU_SEL[2:0] = 111 = Reserved
CLK14	I	CMOS5_0	1	**Oscillator Clock:** Used for 8254 timersHPET (High Precision Event Timer). Runs at 14.31818 MHz. This clock stops during S3 and S5 state.
PCICLK	I	CMOS3_3	1	**PCIClock:** 33 MHz I/O clock. PCICLK provides timing for all transactions on the PCI I/O Bus. ***Note:*** This clock can be stopped in S3 or S5 states. ***Note:*** This signal is not 5 V tolerant. It is 3.3 V tolerant.
CLK48	I	CMOS5_0	1	**48 MHz Clock:** Used to run the USB controller. Runs at 48 MHz. This clock stops during S3 and S5 state.

Table 70. Clocks, Resets, and Miscellaneous Signals (Sheet 2 of 2)

Signal Name	I/O	Class	Pkg Balls	Description
SPKR	O	CMOS3_3	1	**Speaker:** The SPKR signal is the output of counter 2 and is internally "ANDed" with Port 61h bit 1 to provide Speaker Data Enable. This signal drives an external speaker driver device, which in turn drives the system speaker. Upon PLTRST#, its output state is 0. This signal has a weak internal pull-down resistor. ***Note:*** SPKR is sampled at Platform reset as a functional strap. Refer to Table 81 for more details.
RTEST#	I	CMOS3_3	1	**RTC Well Test:** Normally held high (to VccRTC), but can be driven low on the tester or motherboard to test RTC well and resets some bits in the RTC well that are otherwise not reset by PLTRST# or RSMRST#. An external RC circuit on the RTEST# signal creates a time delay such that RTEST# will go high some time after the battery voltage is valid. The RC time delay must be in the 10-20 ms range. This allows detection when a new battery has been installed. Unless entering a XOR Chain test mode, the RTEST# input must always be high when all other non-RTC power planes are on. ***Note:*** This signal is in the RTC power well.
INTVRMEN	I	CMOS3_3	1	**Internal VRM Enable:** This signal is used to enable or disable the integrated 1.5V Voltage Regulators for the Suspend and Auxiliary wells. When connected to Vss, the VRMs are disabled; when connected to the VccRTC power plane, the VRMs are enabled. This signal is in the RTC well. It is not latched and must remain valid for the VRMs to behave properly. Refer to Table 81 for more details.
PLTRST#	O	CMOS3_3	1	**Platform Reset:** The Intel® 3100 Chipset asserts PLTRST# during power-up and when a hard reset sequence is initiated through the CF9h register. PLTRST# is driven inactive a minimum of 1 ms after both PWROK and VGATE are driven high. PLTRST# is driven for a minimum of 1 ms when initiated through the CF9h register. ***Note:*** PLTRST# is in the Resume power plane.
PEB_RPC[1:0]	I/O	CMOS3_3	2	These pins define the PCI Express B port behavior after reset. Refer to the strap section for more details. ***Note:*** PEB_RPC[1:0]# are sampled at platform reset as a functional strap. Refer to Table 81 for more details.
WL_PU0	I	CMOS3_3	1	This pin must be externally pulled high
WL_PU1	I	CMOS3_3	1	This pin must be externally pulled high.
RSTIN#	I	SCHMITT3_3	1	IMCH Reset Input must be connected to PLTRST#
PWRGD	I	SCHMITT3_3	1	IMCH **Power Good:** Asynchronously resets the entire IMCH component, including "sticky" bits. Driven by system logic to indicate all board power supplies are valid.
PE_HPINTR#	I	SCHMITT3_3	1	IMCH **PCI Express Hot-plug Controller Interrupt:** Input pin to hot-plug controller on PCI Express bus. Not 5V tolerant.
Total Signals In Group = 17				

4.4.1.15 LPC and FWH Interfaces

Table 71. LPC and FWH Interfaces

Signal Name	I/O	Class	Pkg Balls	Description
LAD[3:0]/ FWH[3:0]	I/O	CMOS3_3	4	**LPC Multiplexed Command, Address, Data:** Refer to pin state chapter for pull up information. LAD[3:0] may be used as Firmware Hub [3:0] signals.
LFRAME#/ FWH4	O	CMOS3_3	1	**LPC Frame.**:LFRAME# Indicates the start of an LPC cycle, or an abort. LFRAME# may be used as Firmware Hub [4] signal.
LDRQ1#/ GPI41	I	CMOS3_3	1	**LPC Serial EDMA/Master Request Input bit 1:** Used by LPC devices, such as Super I/O chips, to request EDMA or bus master access.This signal is typically connected to external Super I/O device. LDRQ[1]# may optionally be used as GPI[41].
LDRQ0#	I	CMOS3_3	1	**LPC Serial EDMA/Master Request Input bit 0:** Used by LPC devices, such as Super I/O chips, to request EDMA or bus master access.This signal is typically connected to external Super I/O device.
LPCPD#/ SUS_STAT#	O	CMOS3_3	1	**Suspend Status:** This signal is asserted to indicate that the system will be entering a low power state soon. This can be monitored by devices with memory that need to switch from normal refresh to suspend refresh mode. It can also be used by other peripherals as an indication that they must isolate their outputs that may be going to powered-off planes. This signal is called SUS_STAT# on the Power Management Interface.
Total signals in group = 8				

4.4.1.16 USB Interface

Table 72. USB Interface

Signal Name	I/O	Class	Pkg Balls	Description
USBp[3:0], USBn[3:0]	I/O	USB	8	**Universal Serial Bus Port 3Differentials:** These differential pairs are used to transmit Data/Address/Command signals for ports 0, 1, 2 and 3. ***Note:*** External resistors are not required on these signals. The chip integrates the 15 KW pull-down and provides an output driver impedance of 45 Ω which requires no external series resistor.
OC[3:0]#	I		4	**Overcurrent Indicators:** These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred.
USB_RBIASp	I/O	Analog	1	**USB Resistor Bias (Positive):** Analog connection point for an external resistor. Used to set transmit currents and internal load resistors
USB_RBIASn	I/O	Analog	1	**USB Resistor Bias (Negative):** Analog connection point for an external resistor. Used to set transmit currents and internal load resistors

Total signals in group = 14

Notes:

1. The USB signals are all in the RESUME well.
2. All ports support both USB 1.0 and USB 2.0 signaling.
3. OC[3:0]# are not 5 V tolerant. These inputs are only tolerant to 3.3 V specs.

4.4.1.17 SATA Interface

Table 73. SATA Interface

Signal Name	I/O	Class	Pkg Balls	Description
SATA_CLKp, SATA_CLKn	I	SATA	2	**Differential SATA Clock:** 100 MHz clock input from the Clock Generator
SATA_TXp[5:0], SATA_TXn[5:0]	O	SATA	12	**Serial ATA:** Differential Transmit Pair
SATA_RXp[5:0], SATA_RXn[5:0]	I	SATA	12	**Serial ATA:** Differential Receive Pair
SATA_RBIASp	I/0	SATA	1	**Serial ATA Resistor Bias (Positive):** Analog connection point for an external resistor
SATA_RBIASn	I/O	SATA	1	**Serial ATA Resistor Bias (Negative):** Analog connection point for an external resistor
SATA_LED#	O	OD	1	**Serial ATA LED:** This is an open-collector/open-drain output pin driven during SATA command activity. It is to be connected to external circuitry that can provide the current to drive a platform LED. When active, the LED is on. When tristated, the LED is off. An external pull-up resistor is required.
Total signals in group = 35				

4.4.1.18 UART Interface

Table 74. UART Interface (Sheet 1 of 2)

Signal Name	I/O	Class	Pkg Balls	Description
UART_CLK	I	CMOS3_3	1	**UART CLock:** Input clock to the SIU. This clock is passed to the baud clock generation logic for the UART in the SIU.
SIU_RXD[2:1]	I	CMOS3_3	2	**SERIAL INPUT:** Serial data input form device pin to the receive port.
SIU_TXD[2:1]	O	CMOS3_3	2	**SERIAL OUTPUT:** Serial data output to the communication peripheral/modem or data set. Upon reset, the TXD pins will be set to MARKING condition (logic '1' state).
SIU_CTS[2:1]#	I	CMOS3_3	2	**CLEAR TO SEND:** These pins (one for each UART port) are active low, and are used by an external devices signal to the UART if they are ready (or not) to receive data. ***Note:*** This pin could be used as Modem Status Input whose condition can be tested by the processor by reading bit 4 (CTS) of the Modem Status register (MSR). Bit 4 is the compliment of the CTS# signal. Bit0 (DCTS) of the MSR indicates whether the CTS# input has changed state since the previous reading of the MSR. When the CTS bit of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.
SIU_DSR[2:1]#	I	CMOS3_3	2	**DATA SET READY:** Active low, this pin indicates that the external agent is ready to communicate with UARTS. This pin has no effect on the transmitter. ***Note:*** This pin could be used as Modem Status Input whose condition can be tested by the processor by reading bit 5 (DSR) of the Modem status register (MSR). Bit 5 is the complement of the DSR# signal. Bit 1 (DDSR) of the Modem status register (MSR) indicates whether the DSR# input has changed state since the previous reading of the MSR. When the DSR bin of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.

Table 74. UART Interface (Sheet 2 of 2)

Signal Name	I/O	Class	Pkg Balls	Description
SIU_DCD[2:1]#	I	CMOS3_3	2	**DATA CARRIER DETECT:** Active low, this pin indicates that data carrier has been detected by the external agent. ***Note:*** This pin is Modem Status Input which condition can be tested by the processor by reading bit 7 (DCD) of the Modem Status Register (MSR). Bit 7 is complement of the DCD# signal. Bit 3 (DDCD) of the MSR indicates whether the DCD# input has changed state since the previous reading of the MSR. When the DCD bin of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.
SIU_RI[2:1]#	I	CMOS3_3	2	**RING INDICATOR:** Active low, this pin indicates that a telephone ringing signal has been received by the external agent. ***Note:*** This pin is Modem Status Input whose condition can be tested by the processor by reading bit 6 (RI) of the MSR. Bit 6 is the complement of the rI# signal. Bit 2 (TERI) of the MSR indicates whether the RI# input has transitioned back to an inactive state. When the RI bit of the MSR changes from a 1 to 0 an interrupt is generated if the Modem Status Interrupt is enabled.
SIU_DTR[2]#	O	CMOS3_3	2	**DATA TERMINAL READY:** When low, this pin informs the modem or data set that the UART is ready to establish a communication link. The DTR# output signal can be set to an active low by programming the DTR(bit0) of the Modem control register to a logic 1. A Reset operation sets this signal to its inactive state (logic 1). LOOP mode operation holds this signal in its inactive state.
SIU_DTR[1]#	I/O	CMOS3_3	2	**DATA TERMINAL READY:** When low, this pin informs the modem or data set that the UART is ready to establish a communication link. The DTR# output signal can be set to an active low by programming the DTR(bit0) of the Modem control register to a logic 1. A Reset operation sets this signal to its inactive state (logic 1). LOOP mode operation holds this signal in its inactive state. ***Note:*** SIU_DTR[1]# is sampled at Platform reset as a functional strap. Refer to Table 81 for more details.
SIU_RTS[2:1]#	O	CMOS3_3	2	**REQUEST TO SEND:** When low this pin informs the modem or data set that the UART is ready to establish a communication link. The RTS# output signal can be set to an active low by programming the RTS(bit1) of the Modem control register to a logic 1. A Reset operation sets this signal to its inactive state (logic 1). LOOP mode operation holds this signal in its inactive state.
Total signals in group = 17				

4.4.1.19 Interrupt Interface

See Chapter 19.0, "Interrupts," for full behavioral descriptions.

Table 75. Interrupt Interface

Signal Name	I/O	Class	Pkg Balls	Description
SERIRQ	I/O	CMOS3_3	1	**Serial Interrupt Request:** This pin implements the serial interrupt protocol. ***Note:*** SERIRQ, is not 5V tolerant. It is 3.3 V tolerant.
PIRQ[A:D]#	I/O	OD	4	**PCI Interrupt Requests:** The PIRQx# signals can be routed to interrupts 3, 4, 5, 6, 7, 9, 10, 11, 12, 14 or 15 as described in the Interrupt Steering section. Each PIRQx# line has a separate Route Control Register. These signals are 5V tolerant
PIRQ[H:E]#/ GPI[5:2]	I/O	OD	4	**PCI Interrupt Requests:** In Non-APIC mode, the PIRQx# signals are fixed routed These signals are 5 V tolerant. GP[5]/PIRQ[H]# fixed to IRQ[23] GP[4]/PIRQ[G]# fixed to IRQ[22] GP[3]/PIRQ[F]# fixed to IRQ[21] GP[2]/PIRQ[E]# fixed to IRQ[20]
Total signals in group = 9 ***Note:*** The Interrupt Signals, except SERIRQ, are 5 V tolerant. SERIRQ is 3.3 V tolerant.				

4.4.1.20 Power Management Interface

See Chapter 22.0, "Power Management," for full behavioral descriptions.

Table 76. System Management and Power State Signals (Sheet 1 of 2)

Signal Name	I/O	Class	Pkg Balls	Description
THRM#	I	CMOS3_3	1	**Thermal Alarm:** Active low signal generated by external hardware to generate an SMI# or SCI.
THRMTRIP#	I	CMOS1_05	1	**Thermal Trip:** When low, indicates that a thermal trip from the processor occurred, and corrective action will be taken. This input buffer has the same characteristics as the FERR# input buffer.
SLP_S3#	O	CMOS3_3	1	**S3 Sleep Control:** Power plane control. Shuts power to non-critical systems when in the S3 (Suspend To RAM) state.
SLP_S4#	O	CMOS3_3	1	**S4 Sleep Control:** Power plane control. Shuts power to non-critical systems when in the S5 (Soft Off) state. This pin must be used to control the DRAM power in order to use the DRAM power-cycling feature.
SLP_S5#	O	CMOS3_3	1	**S5 Sleep Control:** Power plane control. Shuts power to all non-critical systems when in S5 (Soft Off) states.
SYS_RESET#	I	CMOS3_3	1	**System Reset:** This pin forces a reset after being debounced.
PWROK	I	CMOS3_3	1	**Power OK:** When asserted, PWROK is an indication that core power has been stable for at least 99ms and PCICLK has been toggling cleanly for at least 1 ms. PWROK can be driven asynchronously. When PWROK is low, PLTRST# is asserted. Note that it is required that the core power has been valid for 99ms prior to PWROK assertion. See Table 662 for more details about the PWROK pin functionality.
PWRBTN#	I	CMOS3_3	1	**Power Button:** Causes SMI# or SCI to indicate to system request to go to a sleep state. If already in sleep state, will cause a wake event. If PWRBTN# is pressed for 4 seconds, will cause unconditional transition (power button override) to the S5 state. Override will occur even if the system is in the S3 state.
RI#	I	CMOS3_3	1	**Ring Indicate:** Can be enabled as a wake event and is preserved during power failures.

Table 76. System Management and Power State Signals (Sheet 2 of 2)

Signal Name	I/O	Class	Pkg Balls	Description
RSMRST#	I	CMOS3_3	1	**Resume Well Reset:** Used for resetting the resume well. An external RC circuit is required to guarantee that make sure that the resume well power is valid prior to RSMRST# going high.
SUS_STAT#/ LPCPD#	O	CMOS3_3	1	**Suspend Status:** This signal is asserted to indicate that the system will be entering a low power state soon. This can be monitored by devices with memory that need to switch from normal refresh to suspend refresh mode. It can also be used by other peripherals as an indication that they must isolate their outputs that may be going to powered-off planes. This signal is called LPCPD# on the LPC Interface.
SUSCLK	O	CMOS3_3	1	**Suspend Clock:** Output of the RTC generator circuit (32.768 kHz). SUSCLK will have a duty cycle that can be as low as 30% or as high as 70%.
VRMPWRGD/ VGATE	I	CMOS3_3	1	**Voltage Regulator Power Good:** This is the processor's VRM Power Good, and will save an external AND gate. This signal is internally ANDed with the ATX power supply's PWROK signal. Traditionally, this AND gate has been external to the chipset.
WAKE#	I	CMOS3_3	1	**PCI Express Wake Event:** Sideband wake signal on PCI Express asserted by components requesting wakeup.
PME#	I/O	OD	1	**PCI I/O Power Management Event:** Driven by PCI I/O peripherals to wake the system from low-power states S3 and S5. It can also cause an SCI from the S0 state. Note that in some cases the Intel® 3100 Chipset may drive PME# active (low) due to an internal wake event. It will not drive PME# high (but it may be pulled up using the internal pull-up resistor). ***Notes:*** 1. PME# is in the Resume power plane and has an internal pull-up resistor. 2. This signal is not 5 V tolerant. It is 3.3 V tolerant.
Total signals in group = 15				

4.4.1.21 Watchdog Timer and Real Time Clock Interface

See Chapter 21.0, "Real Time Clock (LPC I/F – D31: F0)," for full behavioral descriptions.

Table 77. Watchdog Timer and Real Time Clock Interfaces

Signal Name	I/O	Class	Pkg Balls	Description
RTCX1	Special	CMOS3_3	1	**Crystal Input 1:** Connected to the 32.768 KHz crystal. If no external crystal is used, then RTCX1 can be driven with the desired clock rate.
RTCX2	Special	CMOS3_3	1	**Crystal Input 2:** Connected to the 32.768 KHz crystal. If no external crystal is used, then RTCX2 should be left floating.
WDT_TOUT#	O	CMOS3_3	1	**Watchdog Timer Output Signal:** The signal is driven low when the main 35-bit down counter reaches zero during the second stage. The WDT_TOUT_CNF bit in the WDT Lock register determines if the output is to change from the previous state if another time out occurs, or WDT_TOUT# is driven low until the system is reset or power is cycled.
Total signals in group = 3 ***Note:*** An external crystal circuit is required for proper operation of the oscillator.				

4.4.1.22 32-bit, 33 MHz PCI Interface

Table 78. PCI Interface (Sheet 1 of 2)

Signal Name	I/O	Class	Pkg Balls	Description
DEVSEL#	I/O	CMOS3_3	1	**Device Select:** The Intel® 3100 Chipset asserts DEVSEL# to claim a PCI transaction. As an output, the Intel® 3100 Chipset asserts DEVSEL# when a PCI master peripheral attempts an access to an internal Intel® 3100 Chipset address or an address destined for NSI (main memory). As an input, DEVSEL# indicates the response to an Intel® 3100 Chipset-initiated transaction on the PCI bus. DEVSEL# is tristated from the leading edge of PCIRST#. DEVSEL# remains tristated by the Intel® 3100 Chipset until driven as a target.
FRAME#	I/O	CMOS3_3	1	**Frame:** FRAME# is driven by the current Initiator to indicate the beginning and duration of an access. While FRAME# is asserted data transfers continue. When FRAME# is negated the transaction is in the final data phase. FRAME# is an input to the Intel® 3100 Chipset when it is the target. FRAME# is an output when the Intel® 3100 Chipset is the initiator.
AD[31:0]	I/O	CMOS3_3	32	**PCI Address/Data:** AD[31:0] signals are multiplexed. During the first clock of a transaction, AD[31:0] contain the physical address (32 bits). After the firs clock, AD[31:0] contain data.
C/BE[3:0]#	I/O	CMOS3_3	4	**Bus Command and Byte Enables:** The command and byte enable signals are multiplexed. During the address phase of a transaction, C/BE[3:0]# define the bus command. During the data phase C/BE[3:0]# are used as Byte Enables. All command encoding not shown are reserved. C/BE[3:0]# — Command Type — Role Supported 0 0 0 0 — Interrupt Acknowledge — None 0 0 0 1 — Special Cycle — None 0 0 1 0 — I/O Read — Target and Initiator 0 0 1 1 — I/O Write 0 1 1 0 — Memory Read 0 1 1 1 — Memory Write 1 0 1 0 — Configuration Read 1 0 1 1 — Configuration Write 1 1 0 0 — Memory Read Multiple 1 1 0 1 — DAC Mode Address 1 1 1 0 — Memory Read Line 1 1 1 1 — Memory Write and Invalidate The Intel® 3100 Chipset will not use reserved values, and will not respond if a PCI master generates a cycle using a reserved value. SeeChapter 15.0, "Device 30, Function 0: PCI to PCI Bridge" for details on how these commands are supported depending on the Intel® 3100 Chipset's role in the PCI cycle (target or initiator).
IRDY#	I/O	CMOS3_3	1	**Initiator Ready:** IRDY# indicates the Intel® 3100 Chipset's ability, as an Initiator, to complete the current data phase of the transaction. It is used in conjunction with TRDY#. A data phase is completed on any clock both IRDY# and TRDY# are sampled asserted. During a write, IRDY# indicates the Intel® 3100 Chipset has valid data present on AD[31:0]. During a read, it indicates the Intel® 3100 Chipset is prepared to latch data. IRDY# is an input to the Intel® 3100 Chipset when it is the target and an output when it is an Initiator.

Table 78. PCI Interface (Sheet 2 of 2)

Signal Name	I/O	Class	Pkg Balls	Description
TRDY#	I/O	CMOS3_3	1	**Target Ready:** TRDY# indicates the Intel® 3100 Chipset's ability to complete the current data phase of the transaction. TRDY# is used in conjunction with IRDY#. A data phase is completed when both TRDY# and IRDY# are sampled asserted. During a read, TRDY# indicates that the Intel® 3100 Chipset, as a Target, has placed valid data on AD[31:0]. During a write, it indicates the Intel® 3100 Chipset, as a Target is prepared to latch data. TRDY# is an input to the Intel® 3100 Chipset when it is the Initiator and an output when it is a Target. TRDY# is tristated from the leading edge of PCIRST#.
STOP#	I/O	CMOS3_3	1	**Stop:** STOP# indicates that Intel® 3100 Chipset, as a target, is requesting an initiator to stop the current transaction. As an Initiator, STOP# causes Intel® 3100 Chipset to stop the current transaction. STOP# is an output when Intel® 3100 Chipset is a Target and an input when Intel® 3100 Chipset is an Initiator.
PAR	I/O	CMOS3_3	1	**Calculated/Checked Parity:** PAR is "even" parity and is calculated on 36 bits – AD[31:0] plus C/BE[3:0]#. "Even" parity means that the number of "1"s within the 36 bits plus PAR are counted and the sum is always even. PAR is always calculated on 36 bits regardless of the valid byte enables. PAR is generated for address and data phases, and is only guaranteed to be valid one PCI clock after the corresponding address or data phase. PAR is driven and tristated identically to the AD[31:0] lines, except that PAR is delayed by exactly one PCI clock. PAR is an output during the address phase (delayed one clock) for all Intel® 3100 Chipset initiated transactions. It is also an output during the data phase (delayed one clock) when the Intel® 3100 Chipset is the initiator of a PCI write transaction, and when it is the Target of a read transaction. The Intel® 3100 Chipset checks parity on the data phase when it is the Initiator of PCI read transactions and when it is the target of PCI write transactions. It also checks parity on the address phase when it is the target of PCI transitions. If a parity error is detected, the Intel® 3100 Chipset will set the appropriate status bits, and has the option to generate an NMI# or SMI#.
PERR#	I/O	CMOS3_3	1	**Parity Error:** Driven by an external PCI device when it receives data that has a parity error. Driven by the Intel® 3100 Chipset when it detects a parity error. The Intel® 3100 Chipset can either generate an NMI# or SMI# upon detecting a parity error (either detected internally or reported via PERR# signal).
REQ[1:0]#	I	CMOS3_3	2	**PCI Requests:** Supports up to 2 external masters on the PCI bus.
GNT[1:0]#	O	CMOS3_3	2	**PCI Grants:** Supports up to 2 masters on the PCI bus.
PLOCK#	I/O	CMOS3_3	1	**PCI Lock:** Same limitations as for 440BX. Indicates an exclusive bus operation and may require multiple transactions to complete. The Intel® 3100 Chipset asserts PLOCK# when it is doing non-exclusive transactions on PCI. PLOCK# is ignored when PCI masters are granted the bus.
SERR#	I/O	OD	1	**System Error:** SERR# can be pulsed active by any PCI device that detects a system error condition. Upon sampling SERR# active, the Intel® 3100 Chipset can be programmed to generate an NMI or SMI#. Implemented as I/O open drain. This allows the Intel® 3100 Chipset to drive these signals.
PCIRST#	O	CMOS3_3	1	**PCI Reset:** This is the secondary PCI Bus reset signal. It is a logical OR of the primary interface PLTRST# signal and the state of the Secondary Bus Reset bit of the Bridge Control register. ***Note:*** PCIRST# is in the resume well.
Total Signals In Group = 50 ***Note:*** PCI signals are 5 V tolerant, except PME# (see Table 76) and PCICLK (see Table 70).				

4.4.1.23 General Purpose I/O Interface

Table 79. General Purpose I/O Interface (Sheet 1 of 2)

Signal Name	I/O	Class	Pkg Balls	Description
GPI[1:0]	I	CMOS5_0	2	**General Purpose Inputs:** Reside in the core power well. Refer to the GPIO chapter to enable GPIO functionality.
GPI[5:2]/ PIRQ[H:E]#	I	OD	4	**General Purpose Inputs:** Reside in the core power well. Can instead be used as PIRQ[H:E]#.
GPI6	I	CMOS3_3	1	**General Purpose Input:** Resides in the core power well. This is not a 5v tolerant buffer. This pin is not available if SMI mode is enabled.
GPI7	I	CMOS3_3	1	**General Purpose Input:** Resides in the core power well. This is not a 5v tolerant buffer.
GPI8	I	CMOS3_3	1	**General Purpose Input:** Resides in the resume power well.
GPI[10:9]	I	CMOS3_3	2	**General Purpose Inputs:** Resides in the resume power well. Refer to the GPIO chapter to enable GPIO functionality.
GPI11/ SMBALERT#	I	CMOS3_3	1	**General Purpose Input:** Resides in the core power well. Can also be used as SMBALERT#.
GPI12	I	CMOS3_3	1	**General Purpose Input:** Resides in the core power well. This is not a 5v tolerant buffer.
GPI13	I	CMOS3_3	1	**General Purpose Input:** Resides in core power well.
GPI[15:14]	I	CMOS3_3	2	**General Purpose Inputs:** Resides in the resume power well. Refer to the GPIO chapter to enable GPIO functionality.
GPO[17:16]	O	CMOS3_3	2	***Note:*** **General Purpose Outputs:** Resides in the core power well. Refer to the GPIO chapter to enable GPIO functionality.GPO[17:16] are sampled at platform reset as a functional strap. Refer to Table 81 for more details.
GPO[19:18]	O	CMOS3_3	2	**General Purpose Outputs:** Resides in the core power well.
GPO20	OD	CMOS3_3	1	**General Purpose Output:** Resides in the core power well.
GPO21	OD	CMOS3_3	1	**General Purpose Output:** Resides in the core power well.
GPO23	OD	CMOS3_3	1	**General Purpose Output:** Resides in the core power well.
GPIO[25:24]	I/O	CMOS3_3	2	***Note:*** **General Purpose Input/Outputs:** Resides in the resume power well.
GPI26	I	CMOS3_3	1	**General Purpose Input:** Resides in the core power well. This is not a 5v tolerant buffer.
GPIO[28:27]	I/O	CMOS3_3	2	**General Purpose Input/Outputs:** Resides in the resume power well.
GPI29	I	CMOS3_3	1	**General Purpose Input:** Resides in the core power well. This is not a 5v tolerant buffer.
GPI30	I	CMOS3_3	1	**General Purpose Input:** Resides in the core power well. This is not a 5v tolerant buffer.
GPI31	I	CMOS3_3	1	**General Purpose Input:** Resides in the core power well. This is not a 5v tolerant buffer.
GPIO32	I/OD	CMOS3_3	1	**General Purpose Input/Output:** Resides in the core power well.
GPIO[34:33]	I/O	CMOS3_3	2	**General Purpose Input/Outputs:** Resides in the core power well.

Table 79. General Purpose I/O Interface (Sheet 2 of 2)

Signal Name	I/O	Class	Pkg Balls	Description
GPI40	I	CMOS5_0	1	**General Purpose Input:** Resides in the core power well. Refer to the GPIO chapter to enable GPIO functionality.
GPI41 LDRQ1#	I	CMOS3_3	1	**General Purpose Input:** Resides in the core power well. Can be used as LDRQ1# as well. This is not a 5 V tolerant buffer.
GPO48	O	CMOS3_3	1	**General Purpose Output:** Resides in the core power well. Refer to the GPIO chapter to enable GPIO functionality.
GPO49/ CPUPWRGD	O	OD	1	**General Purpose Output:** Resides in the core power well. Can be used as CPUPWRGD as well. The external PU resistor tied to this pin shall not be tied to a voltage higher than 1.5 V.
Total Signals in Group =38				

4.4.1.24 Debug Interface

The Debug Interface tables include the TAP and Debug interface signals.

Table 80. TAP and Debug Interface

Signal Name	I/O	Class	Pkg Balls	Description
TRST#	I	SCHMITT1_5	1	Tap Reset Internal PU, 8 k to 80 k, typically 15 k ohms.
TMS	I	SCHMITT1_5	1	Tap Mode Select Internal PU, 8 k to 80 k, typically 15 k ohms.
TDI	I	SCHMITT1_5	1	Tap Serial Data Input Internal PU, 8 k to 80 k, typically 15 k ohms.
TCK	I	SCHMITT1_5	1	Tap Clock Internal PU, 8 k to 80 k, typically 15 k ohms.
TDO	O	OD	1	Tap Serial Data Output
TEST#	I	SCHMITT1_5	1	Test Mode Select Pin: Enables Extended Debug Port Interface pins (DEBUG[7:0]) Internal PU, 8 k to 80 k, typically 15 k ohms.
DEBUG[7:0]	I/O	CMOS1_5	8	XDP (Extended Debug Port) Interface Pins Internal PU, 8 k to 80 k, typically 15 k ohms.
Total Signals in group = 14				

4.5 Pin Straps

The following signals are used for static configuration. They are sampled at reset to select configurations and then revert later to their standard usage once they are changed back to their normal operation.

Note: To invoke the associated mode, the signal must be driven low at least four PCI clocks prior to the time it is sampled.

Table 81. Strapping Information

Purpose	Pin	When Sampled	Polarity	Pull-Up/ Pull-Down	When Intel® 3100 Chipset Starts Driving	Where Readable
No Reboot	SPKR	Rising edge of PWROK	No reboot if sampled high	Weak internal pull-down	When PLTRST# goes high	Memory-mapped Configuration Space, offset 3410h, bit 5
A16 swap override	GPO[16]	Rising edge of PWROK	A16 override if sampled low	Internal Pull-up (15 K-35 K) during PCIRST#	When PLTRST# goes high.	Memory-mapped Configuration Space, offset 3414h, bit 0
Boot BIOS Destination Selection	GPO[17]	Rising edge of PWROK	Destination is LPC when sampled high; Low- PCI	Internal Pull-up (15 K-35 K) during PCIRST#	When PLTRST# goes high	Memory-mapped Configuration Space, offset 3410h, bit 3
Integrated 1.5V Suspend VRM Enable/Disable	INTVRMEN	Always	Integrated VRMs are enabled when high	None	Never	Not Readable
SIW Configuration Port Address Selection	SIU_DTR[1]#	Rising edge of PWROK	SIW Configuration Port Address fixed at 4E/4Fh when sampled high; Low- Port Address at 20E/20Fh	Weak internal pull-up	When PLTRST# goes high	Not Readable
PCI Express B Root Port Configuration	PEB_RPC[1:0]	Rising edge of PWROK	See RPC register description in the Intel® 3100 Chipset configuration	Internal Pull-down (9 K - 50 K) during reset	When PLTRST# goes high	Not Readable

5.0 System Address Map

A system based on the Intel® 3100 Chipset supports up to 16 GBytes of host-addressable memory space and 64 Kbyte+3 of host-addressable I/O space. The IMCH supports full 36-bit addressing for both CPU and I/O subsystem initiated memory space accesses, providing 16 GBytes of addressable space. The I/O and memory spaces are divided by system configuration software into non-overlapping regions. The memory ranges are useful either as system memory or as specialized memory, while the I/O regions are used solely to control the operation of devices in the system.

5.1 Memory Map

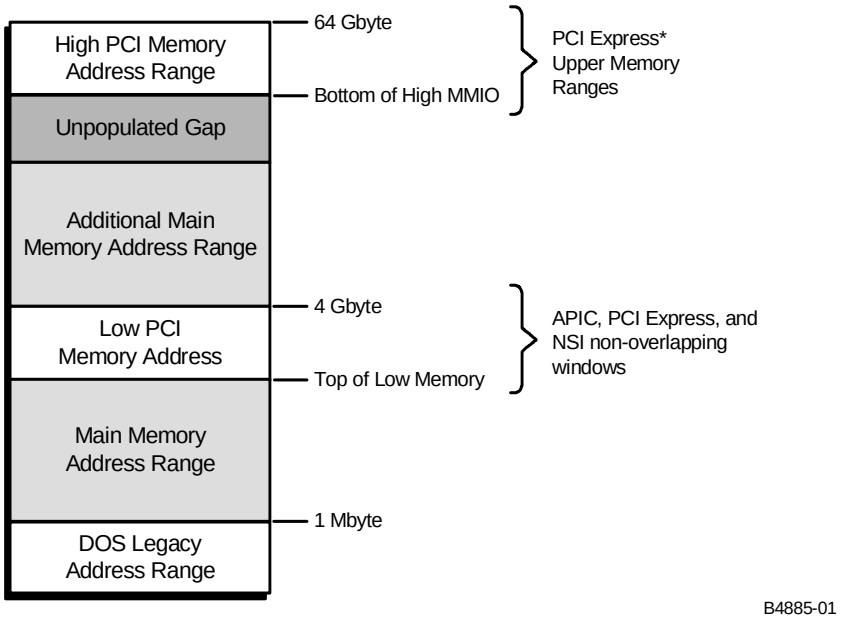
There are five basic regions of memory in the system. They are shown in Table 82. Figure 40 illustrates the basic memory regions.

Table 82. Regions of Memory Ranges

Range	Description
Between top of main memory and 64 GBytes	High PCI Memory Range
Between 4 GBytes and top of main memory	Additional Main Memory Address Range
Between TOLM Register and top of main memory	Low PCI Memory Address Range
Between 1 MByte and the TOLM Register	Main Memory Address Range
Below 1 MByte	DOS Legacy Address Range

Note: The DRAM that physically overlaps the low PCI Memory Address Range (between TOLM and the 4 GByte boundary) may be recovered for use by the system. For example if there is 4 GBytes of physical DRAM and 1 GByte of PCI space, then the system can address a total of 5 GBytes. In this instance, the top GByte of physical DRAM physically located from 3 GBytes to 4 GBytes is addressed between 4 and 5 GBytes by the system.

Figure 40. Basic Memory Regions



5.1.1 System Memory Spaces

Table 83. System Memory Space

	From	To
DOSMEM	0_0000_0000	0_0009_FFFF
MEM1_15	0_0010_0000	0_00EF_FFFF
MAINMEM	0_0100_0000	TOLM
HIGHMEM	1_0000_0000	7_FFFF_FFFF

The address ranges in Table 83 are always mapped to system memory, regardless of the system configuration. The Top of Low Memory (TOLM) register (see Section 13.1.1.58, "Offset C4 - C5h: TOLM – Top of Low Memory Register") provides a mechanism to carve memory out of the MAINMEM segment for use by System Management Mode (SMM) hardware and software, PCI add-in devices, and other functions. The address of the highest 128 MByte quantity of populated DRAM memory in the system is placed into the DRB7 register, which will match the value in the Top of Memory (TOM) register (see Section 13.1.1.62, "Offset CC - CDh: TOM – Top Of Memory Register".

For systems with DRAM space and low PCI memory-mapped space totaling 4 GBytes or less, the value in TOM will match that of the TOLM register. For other memory configurations, the two are unlikely to be the same, since the PCI configuration portion of the BIOS software will program the TOLM register to the maximum value that is less than 4 GBytes and also allows enough room for the total memory space below 4 GBytes (LoPCI) allocated to populated PCI devices.

5.1.2 VGA and MDA Memory Spaces

Table 84 lists the VGA and MDA Memory spaces. Figure 41 illustrates the DOS legacy Region.

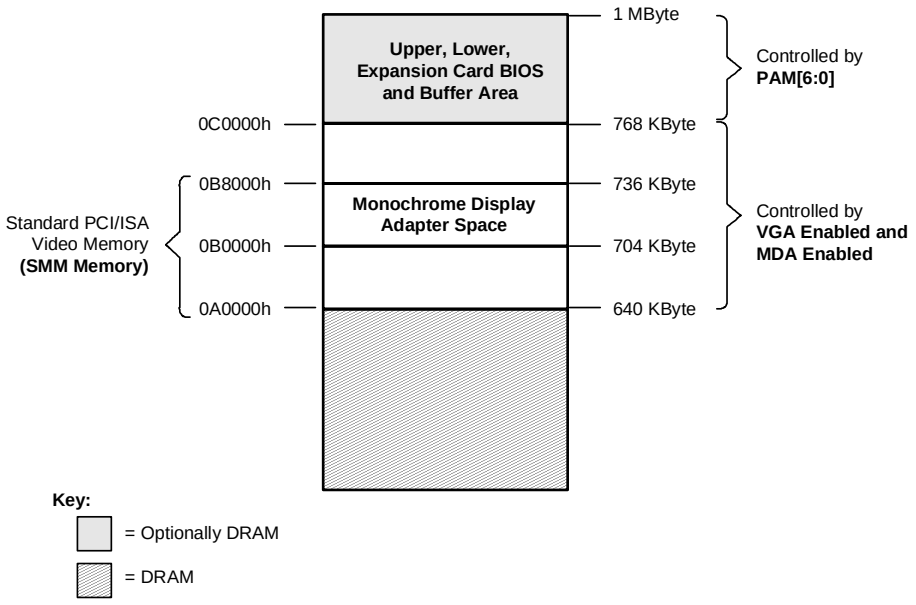
Table 84. IMCH VGA and MDA Memory Spaces

	From	To
VGAA	0_000A_0000	0_000A_FFFF
MDA	0_000B_0000	0_000B_7FFF
VGAB	0_000B_8000	0_000B_FFFF

These legacy address ranges are used on behalf of video cards to map a frame buffer or a character-based video buffer into a dedicated location. By default, accesses to these ranges are forwarded to the NSI. However, if the VGAEN bit is set in one of the BCTRL configuration registers (see Section 13.4.1.25, "Offset 3Eh: BCTRL – Bridge Control Register"), then transactions within the VGA and MDA spaces are sent to one of the PCI Express* interfaces in IMCH.

Note: The VGAEN bit may be set in *one and only one* of the BCTRL registers. Software *must not* set more than one VGAEN bit.

Figure 41. DOS Legacy Region



If the configuration bit EXSMRC.MDAP (see Section 13.1.1.50, "Offset 9Dh: EXSMRC – Extended System Management RAM Control Register") is set, then accesses that fall within the MDA range are sent to NSI without regard for the VGAEN bits. Legacy support requires the ability to have a second graphics controller (monochrome display adapter) in the system. In a Intel® 3100 Chipset system with PCI graphics installed,

accesses in the standard VGA range may be forwarded to any of the logical PCI Express ports (depending on configuration bits). Since the monochrome adapter may be on the NSI (or logical ISA) bus, the IMCH must decode cycles in the MDA range and forward them to NSI. This capability is controlled via the MDAP configuration bit. In addition to the memory range B0000h to B7FFFh, the IMCH decodes I/O cycles at 3B4h, 3B5h, 3B8h, 3B9h, 3BAh and 3BFh and forwards them to NSI.

An optimization allows the system to reclaim the memory displaced by these regions. If SMM memory space is enabled by EXSMRC.G_SMRAME and either the SMRAM.D_OPEN bit (see Section 13.1.1.50, "Offset 9Dh: EXSMRC – Extended System Management RAM Control Register" and Section 13.1.1.51, "Offset 9Eh: SMRAM – System Management RAM Control Register") is set or the processor bus receives an SMM-encoded request for code (not data), then the transaction is steered to system memory rather than NSI. Under these conditions, both the VGAEN bits and the MDAP bit are overridden.

If any VGAEN bit is set, then all ISAEN bits (see Section 13.4.1.25, "Offset 3Eh: BCTRL – Bridge Control Register") must be set. The PCI Specification defines VGAEN to be 10-bit decode. Therefore the other peer bridges must also be 10-bit decodes (ISAEN), so that two or more devices don't claim same access. Bridge C doesn't know bridge B has its VGAEN bit set.

The MDA bit may only be set when one of the VGAEN bits is set. If no VGAEN bit is set, then MDA must not be set either. When the VGA range is already mapped onto the NSI interface, the MDA range is included as a subset, and the MDA enable is meaningless.

5.1.3 PAM Memory Spaces

The Address range for the PAM memory space is defined in Table 85.

Table 85. IMCH PAM Memory Address Ranges

	From	To
PAMC0	0_000C_0000	0_000C_3FFF
PAMC4	0_000C_4000	0_000C_7FFF
PAMC8	0_000C_8000	0_000C_BFFF
PAMCC	0_000C_C000	0_000C_FFFF
PAMD0	0_000D_0000	0_000D_3FFF
PAMD4	0_000D_4000	0_000D_7FFF
PAMD8	0_000D_8000	0_000D_BFFF
PAMDC	0_000D_C000	0_000D_FFFF
PAME0	0_000E_0000	0_000E_3FFF
PAME4	0_000E_4000	0_000E_7FFF
PAME8	0_000E_8000	0_000E_BFFF
PAMEC	0_000E_C000	0_000E_FFFF
PAMF0	0_000F_0000	0_000F_FFFF

The 256 Kbyte Programmable Access Memory (PAM) region is divided into three parts:

- ISA expansion region, a 128 Kbyte area between 0_000C_0000h – 0_000D_FFFFh
- Extended BIOS region, a 64 Kbyte area between 0_000E_0000h – 0_000E_FFFFh
- System BIOS region, a 64 Kbyte area between 0_000F_0000h – 0_000F_FFFFh.

Specialized programmable hardware in the IMCH supports routing of read and write accesses within the PAM region independently to memory or to NSI.

Non-snooped transactions are treated accordingly:

- Non-snoop Reads: Memory address 0h. The result is an unsupported request (UR) completion.
- Non-snoop Writes: Memory address 0h with byte enables deasserted.

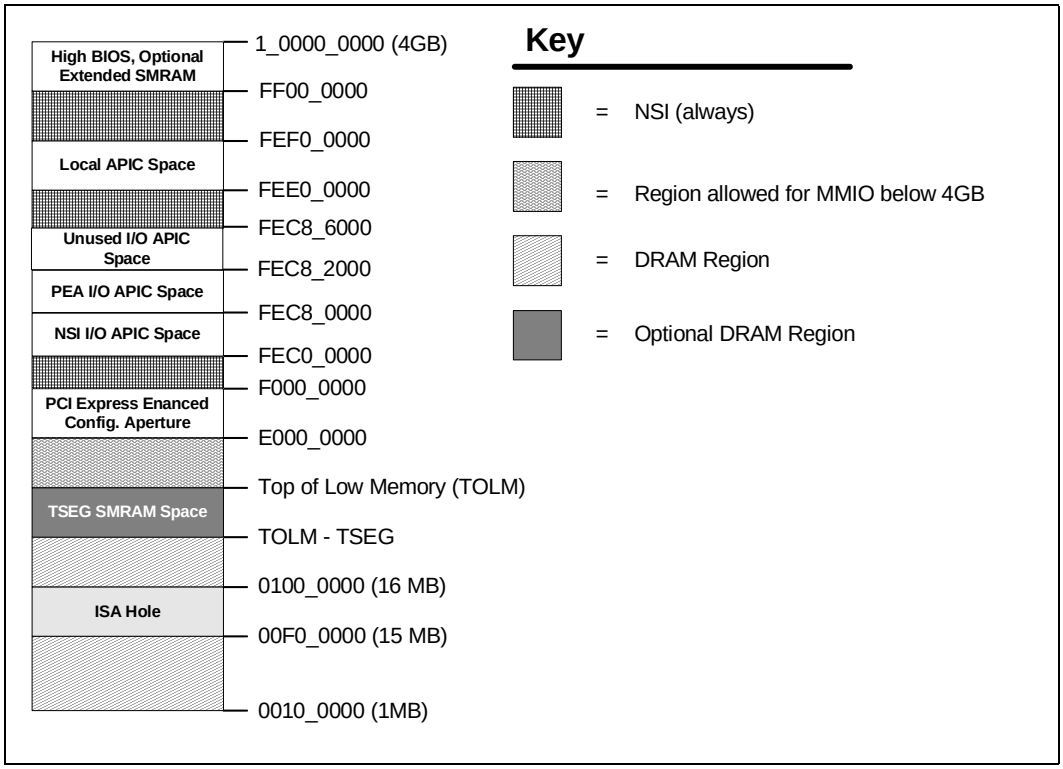
The ISA expansion region is divided into eight 16 Kbyte segments. Each segment can be assigned one of four Read/Write memory states: read-only, write-only, read/write, or disabled. These segments are typically set to disabled for memory access, which leaves them routed to NSI for ISA space.

The extended System BIOS region is divided into four 16 Kbyte segments. Each segment can be assigned independent read and write attributes so it can be mapped either to main DRAM or to NSI. Typically, this area is used for RAM or ROM.

The system BIOS region is a single 64 Kbyte segment. This segment can be assigned independent memory read and write attributes. It is by default (after reset) Read/Write disabled and cycles are forwarded to NSI. By manipulating the Read/Write attributes, the IMCH can “shadow” BIOS into the main DRAM. The term “shadow” is used to describe the condition where ROM memory has been duplicated into main memory; such that reads are serviced from memory, while writes are directed back to the original ROM device. Such a configuration allows low-latency reads of BIOS information from the ROM while preventing malicious or inadvertent alteration of the BIOS information in use.

Note: The PAM regions are generally inaccessible from the logical PCI Express ports. All inbound writes from any port that hit the PAM regions are sent to NSI, which prevents corruption of non-volatile data shadowed in main memory. All inbound reads from any port that hit the PAM regions are harmlessly terminated internally; data is returned, but not necessarily from the requested address. Transaction routing is not hardware enforced based on the settings in the PAM configuration registers.

Figure 42. Memory Region from 1 MByte through 4 GBytes



The IMCH allows programmable memory attributes on 13 legacy memory segments of various sizes in the 768 Kbyte to 1 Mbyte (C0000h – FFFFFh) and 640 Kbytes to 1 Mbyte address range. Seven Programmable Attribute Map (PAM) registers are used to support these features. Not all seven of these registers are identical. PAM0 controls only one segment (high), while PAM[1:6] each control two segments (high and low). Cache ability of these areas is controlled via the MTRR registers in the processor. The following two bits apply to both host accesses and PCI initiator accesses to the PAM areas and are used to specify the memory attributes for each memory segment. These bits apply to both host accesses and PCI initiator accesses to the PAM areas.

RE Read Enable. When RE = 1, the processor read accesses to the corresponding memory segment are claimed by the IMCH and directed to main memory. Conversely, when RE = 0, the host read accesses are directed to the IICH's PCI bus.

WE Write Enable. When WE = 1, the host write accesses to the corresponding memory segment are claimed by the IMCH and directed to main memory. Conversely, when WE = 0, the host write accesses are directed to the IICH's PCI bus.

Together, these two bits specify memory attributes (Read-Only, Write Only, Read/Write and Disabled) for each memory segment. These bits only apply to host-initiated access to the PAM areas. The IMCH forwards to main memory any PCI Express initiated accesses to the PAM areas. At the time such PCI Express accesses to the PAM region may occur, the targeted PAM segment must be programmed to Read/Write. It is illegal to issue a PCI Express initiated transaction to a PAM region with the associated PAM register not set to Read/Write.

As an example, consider a BIOS that is implemented on the expansion bus. During the initialization process, BIOS can be shadowed to main memory to increase system performance. When BIOS is shadowed to main memory it must be copied to the same address location. To shadow the BIOS, the attributes for that address range must be set to Write-Only. The BIOS is shadowed by first doing a read of that address, which is forwarded to the expansion bus. The host then writes the same address, which is directed to main memory. After BIOS is completely shadowed, the attributes for that memory area are changed to Read-Only so that all writes are forwarded to the expansion bus. Figure 43 and Table 86 show the PAM registers and the associated attribute bits.

Figure 43. PAM Associated Attribute Bits

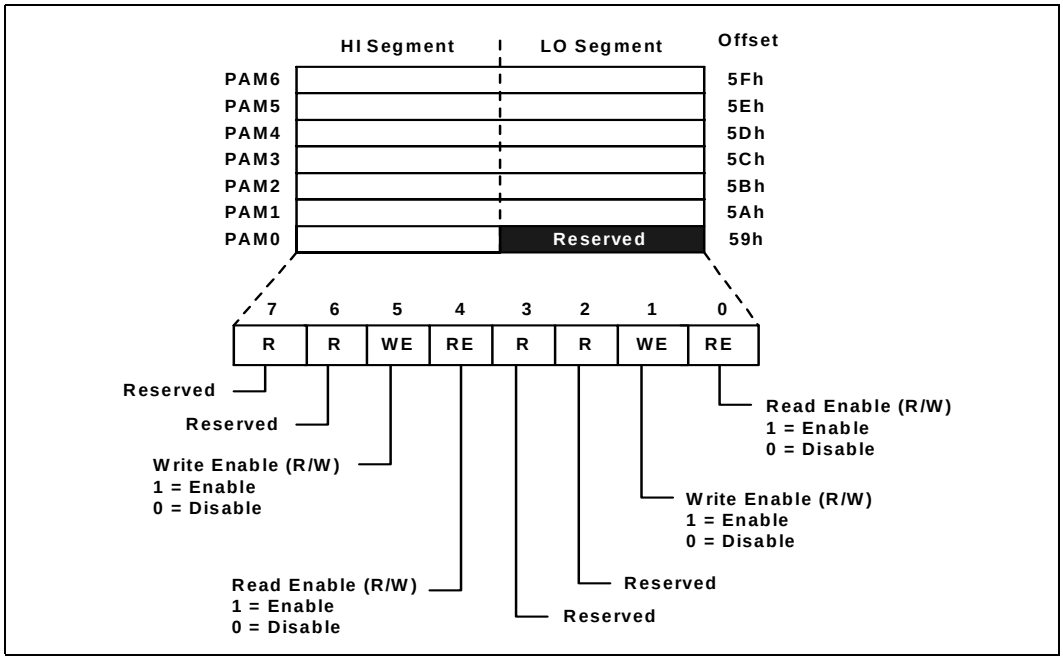


Table 86. PAM Associated Attribute Bits

PAM Reg	Attribute Bits		Memory Segment	Comments	D0:F0 Offset
PAM0 03:00, 07:06	Reserved		—	Reserved	59h
PAM0 05:04	WE	RE	0F0000h–0FFFFFh	BIOS Area	59h
PAM1 03:02, 07:06	Reserved		—	Reserved	5Ah
PAM1 01:00	WE	RE	0C0000h–0C3FFFh	BIOS Area	5Ah
PAM1 05:04	WE	RE	0C4000h–0C7FFFh	BIOS Area	5Ah
PAM2 03:02, 07:06	Reserved		—	Reserved	5Bh
PAM2 01:00	WE	RE	0C8000h–0CBFFFh	BIOS Area	5Bh
PAM2 05:04	WE	RE	0CC000h–0CFFFFh	BIOS Area	5Bh
PAM3 03:02, 07:06	Reserved		—	Reserved	5Ch
PAM3 01:00	WE	RE	0D0000h–0D3FFFh	BIOS Area	5Ch
PAM3 05:04	WE	RE	0D4000h–0D7FFFh	BIOS Area	5Ch
PAM4 03:02, 07:06	Reserved		—	Reserved	5Dh
PAM4 01:00	WE	RE	0D8000h–0DBFFFh	BIOS Area	5Dh
PAM4 05:04	WE	RE	0DC000h–0DFFFFh	BIOS Area	5Dh
PAM5 03:02, 07:06	Reserved		—	Reserved	5Eh
PAM5 01:00	WE	RE	0E0000h–0E3FFFh	BIOS Extension	5Eh
PAM5 05:04	WE	RE	0E4000h–0E7FFFh	BIOS Extension	5Eh
PAM6 03:02, 07:06	Reserved		—	Reserved	5Fh
PAM6 01:00	WE	RE	0E8000h–0EBFFFh	BIOS Extension	5Fh
PAM6 05:04	WE	RE	0EC000h–0EFFFFh	BIOS Extension	5Fh

See Section 13.1.1.19, “Offset 59h: PAM0 – Programmable Attribute Map 0 Register” through Section 13.1.1.25, “Offset 5Fh: PAM6 – Programmable Attribute Map 6 Register” for more register information on PAM memory space registers.

5.1.4 ISA Hole Memory Space

Table 87. ISA Hole Memory Space

	From	To
ISA15	0_00F0_0000	0_00FF_FFFF

BIOS software may optionally open a “window” between 15 MBytes and 16 MBytes that relays transactions to NSI instead of completing them with a system memory access. This window is opened with the FDHC.HEN configuration field. See Section 13.1.1.18, “Offset 58h: FDHC – Fixed DRAM Hole Control Register” for more details on this register.

5.1.5 TSEG SMM Memory Space

Table 88. TSEG SMM Memory Space

	From	To
TSEGSMM	TOLM - TSEG	TOLM

The TSEG SMM space allows system management software to partition a region of main memory just below the top of low memory (TOLM) that is accessible only by system management software.

Size — 128 kB, 256 kB, 512 kB, or 1 MByte in size, depending upon the EXSMRC.TSEG_SZ field (see Section 13.1.1.50).This space must be below 4 GBytes, so it is specified relative to TOLM and not relative to the top of physical memory.

Enabling — SMM memory is globally enabled by EXSMRC.G_SMRAME (see Section 13.1.1.50). Requests may access SMM system memory when either SMM space is open (see SMRAM.D_OPEN in Section 13.1.1.51) or the IMCH receives an SMM code request on its processor bus.

Access — In order to access the TSEG SMM space, the TSEG must be enabled by EXSMRC.T_EN (Section 13.1.1.50). When all of these conditions are met, then a processor bus access to the TSEG space (between TOLM-TSEG and TOLM) is sent to system memory. If the high SMRAM is not enabled or if the TSEG is not enabled, then all memory requests from all interfaces are forwarded to system memory. If the TSEG SMM space is enabled, and an agent attempts a non-SMM access to TSEG space, then the transaction is specially terminated.

Inbound accesses from NSI or PCI Express ports are not allowed to access SMM space.

5.1.6 PCI Express* Enhanced Configuration Aperture

Table 89. PCI Express* Enhanced Configuration Aperture

	From	To
HECREGION	0_E000_0000	0_EFFF_FFFF

PCI Express defines a memory-mapped aperture mechanism through which to access 4 Kbyte of PCI configuration register space for each possible bus, device, and function number. This 4 Kbyte space includes the compatible 256 B of register offsets that are traditionally accessed via the legacy CF8/CFC configuration aperture mechanism in I/O address space, making the enhanced configuration mechanism a full superset of the legacy mechanism. The enhanced mechanism has the advantage that full destination and type of access is specified in a single memory-mapped uncacheable transaction on the FSB, which is both faster and more robust than the historical I/O-mapped address and data register access pair.

The Intel® 3100 Chipset places the enhanced configuration aperture at E000_0000h by default, as this is the first contiguous 256 MByte location below the 4 GByte boundary available for such usage.

The Intel® 3100 Chipset provides for relocation of this aperture via the HECBASE register (see Section 13.1.1.63, "Offset CE - CFh: HECBASE – PCI Express Port A (PEA) Enhanced Configuration Base Address Register", although validation of moving the region is minimal.

5.1.7 IOAPIC Memory Space

Table 90. IOAPIC Memory Space

	From	To
IOAPIC0 (NSI)	0_FEC0_0000	0_FEC7_FFFF
IOAPIC2 (PEA0)	0_FEC8_0000	0_FEC8_0FFF
IOAPIC3 (PEA1)	0_FEC8_1000	0_FEC8_1FFF

The IOAPIC spaces are used to communicate with IOAPIC interrupt controllers that may be populated on NSI through PCI Express port A (PEA). Since it is difficult to relocate an interrupt controller using plug-and-play software, fixed address decode regions have been allocated for them. Processor accesses to the IOAPIC0 region are always sent to NSI. Processor accesses to the IOAPIC2 region are always sent to PEA. These regions are subject to the APIC disable, which are cleared by BIOS after the allocated regions have been reflected down to the base registers of APIC controllers discovered during standard enumeration. Until this step of the initialization sequence has been performed, accesses to these regions are treated as subtractive decode and routed to NSI.

The IMCH does not support an IOAPIC range for the EDMA controller, since there is no IOxAPIC device or corresponding register set integrated into the EDMA controller.

5.1.8 FSB Interrupt Memory Space

Table 91. FSB Interrupt Memory Space

	From	To
FSBINTR	0_FEE0_0000	0_FEEF_FFFF

The FSB Interrupt space is the address range used to deliver interrupts to the FSB. Any device below NSI or a PCI Express port may issue a Memory Write to 0FEEx_xxxxh. The IMCH will forward this Memory Write along with its associated data to the FSB as a Message Signaled Interrupt (MSI) transaction. The IMCH terminates the FSB transaction by asserting TRDY# and providing the response. This Memory Write cycle does not go to DRAM.

The processors may also use this region to send inter-processor interrupts (IPI) from one processor to another. IMCH support for this feature includes the ability to handle redirect-able MSI transactions according to values programmed into integrated task priority registers.

Reads to this address range are aborted by the IMCH.

5.1.9 High SMM Memory Space

Table 92. High SMM Memory Space

	From	To
HIGHSMM	0_FEDA_0000	0_FEDB_FFFF

The HIGHSMM space allows cacheable access to the compatible SMM space by remapping valid SMM accesses between 0_FEDA_0000 and 0_FEDB_FFFF to physical accesses between 0_000A_0000 and 0_000B_FFFF. The accesses are remapped when SMRAM space is enabled, an appropriate access is detected on the processor bus, and when EXSMRC.H_SMRAME (Section 13.1.1.50) allows access to high SMRAM space. Inbound SMM memory accesses from any port are specially terminated; reads are provided with data retrieved from address 0, while writes are ignored entirely (all byte enables deasserted).

5.1.10 PCI Device Memory (MMIO)

The IMCH provides two distinct regions of memory that may be mapped to populated PCI devices. The first is the traditional (non-prefetchable) MMIO range, which must lie below the 4 GByte boundary. The registers associated with non-prefetchable MMIO (MBASE/MLIMIT, see Section 13.4.1.16, "Offset 20 - 21h: MBASE – Memory Base Address Register"/Section 13.4.1.17, "Offset 22 - 23h: MLIMIT – Memory Limit Address Register") are unchanged from historical 32-bit architecture IMCH implementations. The second is the prefetchable MMIO range, which has been extended in the Intel® 3100 Chipset such that it may lie on either side of the 4 GByte boundary. The registers associated with prefetchable MMIO (PMBASE/PMLIMIT, see Section 13.4.1.18, "Offset 24 - 25h: PMBASE – Prefetchable Memory Base Address Register"/Section 13.4.1.19, "Offset 26 - 27h: PMLIMIT – Prefetchable Memory Limit Address Register") have been augmented by the PCI defined upper 32-bit base/limit register pair (PMBASU/PMLMTU, see Section 13.4.1.20, "Offset 28h: PMBASU – Prefetchable Memory Base Upper Address Register"/Section 13.4.1.21, "Offset 2Ch: PMLMTU – Prefetchable Memory Limit Upper Address Register"), although only the first nibble of each register is implemented in the IMCH.

The MBASE/MLIMIT pair must be programmed to lie between TOLM and 4 GBytes. The PMBASE/PMLIMIT and PMBASU/PMLMTU registers must be programmed to lie between TOLM and 4 GBytes.

Because these registers define a PCI memory space, they are subject to the memory access enable (MAE) control bit in the standard PCI command register (see Section 13.4.1.3, "Offset 04 - 05h: PCICMD – PCI Command Register").

Note: Using the same address space as both cacheable and non cacheable is discouraged. Also, assigning and writing the same host address space to two independent downstream devices is also discouraged. Although not illegal, both of the above conditions are very difficult to setup intelligently and validate. If two devices decide to use the same memory space, and they both send write cycles to it (both either cacheable or uncacheable), there are no guarantees that device 1 data (being older) will get there before device 2 data (being newer) if they do not use a flagging mechanism.

5.1.10.1 Device 2 Memory and Prefetchable Memory

Table 93. Device 2 Memory and Prefetchable Memory

	From	To
M2	MBASE2	MLIMIT2
PM2	PMBASE2/PMBASU2	PMLIMIT2/PMLMTU2

Plug-and-play software configures the PEA a memory window in order to provide enough memory space for the devices behind this virtual PCI-to-PCI bridge. Accesses whose addresses fall within these windows are decoded and forwarded to PEA0 for completion. Note that neither region should overlap with any other fixed or relocate-able area of memory. Also note that PCICMD2 refers to PCICMD for device 2.

5.1.10.2 Device 3 Memory and Prefetchable Memory

Table 94. Device 3 Memory and Prefetchable Memory

	From	To
M3	MBASE3	MLIMIT3
PM3	PMBASE3/PMBASU3	PMLIMIT3/PMLMTU3

Plug-and-play software configures the PEA1 memory window in order to provide enough memory space for the devices behind this virtual PCI-to-PCI bridge. Accesses whose addresses fall within this window are decoded and forwarded to PEA1 for completion. Note that neither region should overlap with any other fixed or relocate-able area of memory. Also note that PCICMD3 refers to PCICMD for device 3.

If PCI Express Port A0 is configured to operate in x8 mode, all functional space for PEA1 disappears, effectively collapsing M3/PM3 to match the limit addresses of M2/PM2.

5.2 IMCH Responses to EDMA Transactions

In the following tables, the term “Abort” implies that the EDMA engine will immediately stop the transfer in progress. The offending access will not be forwarded to the inbound/outbound arbiter at all, an error bit will be set accordingly, and the error will be escalated as specified by the configuration bits controlling interrupts and errors.

Note: This behavior is quite different from the PCI Express inbound ports, as in the latter cases the transaction in question was requested by some other initiator elsewhere in the platform. The EDMA engine is a source of traffic all by itself, which makes error containment much simpler in the case of EDMA traffic.

5.2.1 Fixed Address Spaces (EDMA)

Table 95 summarizes IMCH responses to EDMA accesses to the various fixed address spaces.

Table 95. EDMA Accesses to Fixed Address Spaces

Address Space	Conditions	Destination	IMCH Response
DOSMEM	-	MainMem	Transaction is sent to memory system
VGAA VGAB MDA PAMC0... PAMF0	-	Abort	Programmer's responsibility not to target EDMA accesses in the legacy region between 640 kB and 1 MByte
MEM1_15	-	MainMem	Transaction is sent to memory system
ISA15	FDHC.HEN = 0	MainMem	Transaction is sent to memory system
	FDHC.HEN = 1	Abort	EDMA will abort on accesses directed to the ISA hole when enabled
MAINMEM	-	MainMem	Transaction is sent to memory system unless address hits an enabled TSEG SMM range. See TSEGSMM.
TSEGSMM	-	Variable	Refer to **Table 97, "Supported SMM Ranges"**
IOAPIC[0,2-3]	-	Abort	Programmer's responsibility to avoid the APIC ranges
FSBINTR	-	Abort	Programmer's responsibility to avoid the FSB interrupt messaging range
HIGHSMM	-	Variable	Refer to **Table 97, "Supported SMM Ranges"**
HIGHMEM	Address is below the top of memory space defined by TOM and the REMAP registers	MainMem	Transaction is sent to memory system
	Address is above the top of memory	Abort	Hardware will detect an attempt to access above the populated DRAM space, and will abort.

5.2.2 Relocatable Address Spaces (EDMA)

Table 96 summarizes IMCH responses to EDMA accesses to the various relocatable address spaces.

Note: EDMA access is not permitted to the port, thus any address mapping to the legacy interface will cause an abort.

Table 96. EDMA Accesses to Relocatable Address Spaces

Address Space	Conditions	Destination	IMCH Response
NSI: M NSI: PM	-	Abort	No support for EDMA destination on NSI.
PEA: M[n] PEA: PM[n]	Write, MAE = 1	PEA[n]	Transaction forwarded to destination PEA port.
	Write, MAE = 0	Abort	Abort. Memory access disabled.
	Read Transaction	Abort	Abort. No support for peer segment reads.
NSI_SUB	-	Abort	No support for EDMA destination on NSI.

Table 96. EDMA Accesses to Relocatable Address Spaces

Address Space	Conditions	Destination	IMCH Response
PEA: PM[n]	Write, MAE = 1	PEA[n]	Transaction forwarded to destination PEA port.
	Write, MAE = 0	Abort	Abort. Memory access disabled.
	Read Transaction	Abort	Abort. No support for peer segment reads.

5.3 I/O Address Space

The IMCH does not support the existence of any I/O devices on the processor bus. The IMCH generates outbound transactions on behalf of all processor I/O accesses. The IMCH contains two internal registers in processor CPU I/O space dedicated to the configuration access mechanism; the Configuration Address Register (CONFIG_ADDRESS) and the Configuration Data Register (CONFIG_DATA). The behavior of the IMCH in response to accesses to these registers is described in Chapter 6.0, "Platform Configuration."

The processor allows 64 K+3 bytes to be addressed within the I/O space. The IMCH propagates the CPU I/O address without any translation to the targeted destination bus. Note that the upper three locations can be accessed only during I/O address wrap-around; when signal A16# is asserted on the processor bus. A16# is asserted on the processor bus whenever a Dword I/O access is made from address 0FFFDh, 0FFFEh, or 0FFFFh. In addition, A16# is asserted when software attempts a two-byte I/O access from address 0FFFFh.

All I/O accesses (read or write) which do not map to internal IMCH registers will receive a DEFER response on the FSB, and be forwarded to the appropriate outbound port. The IMCH never posts an I/O write.

The IMCH never responds to inbound transactions to I/O or configuration space initiated on any port. Inbound I/O or configuration transactions requiring a completion are terminated with "master abort" completion packets on the originating port interface. Inbound I/O or configuration write transactions not requiring completion are dropped.

5.3.1 Configuration Window

The I/O addresses 0CF8h and 0CFCh are treated specially, as they define the compatible configuration window. Dword accesses to 0CF8h address the internal IMCH configuration address register. Accesses from 1 to 4 bytes in size to the region from 0CFC-0CFFh are treated as configuration data accesses if configuration space is enabled (bit31 of the configuration address register is set). Refer to Chapter 6.0, "Platform Configuration" for further details.

5.3.2 VGA and MDA Regions

Along with the memory space address regions described in Section 5.1.2, "VGA and MDA Memory Spaces", there are fixed I/O locations associated with both the VGA and the MDA regions. Accesses to these addresses are routed to NSI by default, but this behavior may be modified via the VGAEN, MDAP and MAE configuration settings. Refer to the prior sections for the rules associated with the configuration settings of VGAEN, MDAP and MAE.

The MDA region includes I/O space addresses 3B4h, 3B5h, 3B8h, 3B9h, 3BAh and 3BFh. The VGA region includes I/O space ranges 3B0-3BBh, and 3C0-3DFh. The *PCI Specification* defines both MDA and VGA to be 10-bit address decode, thus all accesses

with A[9:0] matching any of these addresses are subject to the associated routing rules. Address bits A[15:10] are ignored when the check against these fixed addresses are applied.

The order of precedence for the routing checks is as follows:

- MAE = 0, MDA addresses will route to NSIif MDAP is set, overriding any VGAEN
- MAE = 1, MDA addresses will route to the Peer deviceif MDAP is set, overriding any VGAEN
- MAE = 0, VGA addresses will route to the NSI
- MAE = 1, VGA addresses will route to the PCI Express port with its VGAEN set, if any
- MAE = 0, MDA addresses which fall within VGA regions will route to the NSI if MDAP is clear
- MAE = 1, MDA addresses which fall within VGA regions will follow VGAEN if MDAP is clear
- Both VGA and MDA addresses default to NSI if MDAP and all VGAEN bits are clear

Note: Setting of MDAP or any of the VGAEN bits implies that the ISAEN bit is also set in all virtual P2P bridges, because of the 10-bit decode requirement.

5.4 Main Memory Addressing

The “High Memory” and “Extended Memory” address regions are together called “Main Memory.” Main memory is composed of address segments that refer to SDRAM system memory. Main memory addresses are mapped to SDRAM channels, devices, banks, rows, and columns in different ways depending upon the type of memory being used and upon the density or organization of the memory. The process for determining the device and channel IDs for addressed devices is as follows:

- The requested address is compared against the values of all eight DRB registers. The number of the register whose programmed value is greater than the address and whose previous register is less than the address is the output of the comparison.
- The value of the DRB register “below” is subtracted from the address in order to determine the offset into the group.
- The offset determines the manner in which the row, column, and bank address bits are extracted from the address.

The process of mapping within a row once an offset has been calculated is detailed in Chapter 12.0, “Supported DRAM Technology.”

5.5 System Management Mode (SMM) Space

The Intel® 3100 Chipset supports the use of main memory as System Management RAM (SMM RAM) enabling the use of System Management Mode. The IMCH supports three SMM options:

- Compatible SMRAM (C_SMRAM)
- High Segment (HSEG)
- Top of Memory Segment (TSEG)

System Management RAM space provides an access protected memory area that is available for SMI handler code and data storage. This memory resource is normally hidden from the Operating System so that the processor has immediate access to this memory space upon entry to SMM (cannot be swapped-out).

5.5.1 SMM Addressing Ranges

IMCH provides three SMRAM options:

- Below 1 MByte option that supports compatible SMI handlers
- Above 1 MByte option that allows new SMI handlers to execute with write-back cacheable SMRAM
- Optional larger write-back cacheable T_SEG area from 128 Kbyte to 1 MByte in size. The above 1 MByte solutions require changes to compatible SMRAM handler code to properly execute above 1 MByte

Note: The first two options both map legal accesses to the same physical range of memory, while the third defines an independent region of addresses.

5.5.1.1 SMM Space Restrictions

If any of the following conditions are violated, the results of SMM accesses are unpredictable and may cause the system to hang:

- The Compatible SMM space must not be set-up as cacheable.
- Both D_OPEN and D_CLOSE must not be set to 1 at the same time.
- When TSEG SMM space is enabled, the TSEG space must not be reported to the OS as available DRAM. This is a BIOS responsibility.

BIOS and SMM code must cooperate to properly configure the IMCH in order to ensure reliable operation of the SMM function.

5.5.1.2 SMM Space Definition

SMM space is defined by both its addressed SMM space and its DRAM SMM space. The addressed SMM space is defined as the range of FSB addresses used by the CPU to access SMM space. DRAM SMM space is defined as the range of physical DRAM memory locations containing SMM information.

The SMM space can be accessed at one of three transaction address ranges:

- Compatible
- High
- TSEG

The Compatible and TSEG SMM space is not remapped and therefore the addressed and DRAM SMM physical addresses are identical. The High SMM space is remapped; thus the addressed and DRAM SMM locations are different. Note that the High DRAM space is the same as the Compatible Transaction Address space.

Table 97 describes all three unique addressing combinations:

- Compatible Transaction Address
- High Transaction Address
- TSEG Transaction Address

Table 97. Supported SMM Ranges

SMM Space Enabled	Transaction Address Space (Adr)	DRAM Space (DRAM)
Compatible (C)	A0000h to BFFFFh	A0000h to BFFFFh
High (H)	0FEDA0000h to 0FEDBFFFFh	A0000h to BFFFFh
TSEG (T)	(TOLM-TSEG_SZ) to TOLM	(TOLM-TSEG_SZ) to TOLM

Notes:

1. High SMM: This implementation is consistent with the Intel E7500 and Intel E7501 designs. In prior MCH designs the High segment was the 384 Kbyte region from A_0000h to F_FFFFh. However C_0000h to F_FFFFh was not useful, so it has been deleted in the IMCH design.
2. TSEG SMM: This implementation is consistent with the Intel E7500 and Intel E7501 designs. In prior MCH designs the TSEG address space was offset by 256 MBytes to allow for simpler decoding and the TSEG was remapped to just under the TOLM. In the IMCH the TSEG region is not offset by 256 MBytes and it is not remapped.

5.6 Memory Reclaim Background

The following Memory Mapped I/O devices and ranges are typically located below 4 GBytes:

- High BIOS
- H-Seg
- XAPIC
- Local APIC
- FSB Interrupts
- PEA0 through PEA1 M, PM and BAR regions

In previous generation MCH architectures, the physical DRAM memory overlapped by the logical address space allocated to these Memory Mapped I/O devices was unusable. In server systems the memory allocated to memory mapped I/O devices could easily exceed 1 GByte. This creates the possibility of a large amount of physical memory populated in the system becoming unusable.

The IMCH provides the capability to reclaim the physical memory overlapped by the Memory Mapped I/O logical address space via remapping physical memory from the Top of Low Memory (TOLM) boundary up the 4 GBytes boundary (or TOM if less than 4 GBytes) to an equivalent sized logical address range located just above the top of physical memory.

5.6.1 Memory Remapping Algorithm

Terminology clarification:

Physical Address — The address presented to the IMCH is traditionally called a “physical address,” because Intel architecture processors contain both segmentation and paging hardware, and all compatible software differentiates between logical addresses, virtual addresses, and physical addresses. The algorithm for remapping addresses presented to the IMCH to reclaim DRAM address space must be implemented such that the mechanism is invisible to compatible software.

System Address — The system address applies to the internal IMCH interface to physical DRAM memory, and is not directly visible to software, other than through certain internal logging registers used to store decoded DRAM address information for error isolation.

An incoming address (referred to as a physical address) is checked to see if it falls in the memory remap window. The bottom of the remap window is defined by the value in the REMAPBASE register (see Section 13.1.1.59, "Offset C6 - C7h: REMAPBASE – Remap Base Address Register"). The top of the remap window is defined by the value in the REMAPLIMIT register (Section 13.1.1.60, "Offset: C8 - C9h: REMAPLIMIT – Remap Limit Address Register"). An address that falls within this window is remapped to the physical memory starting at the address defined by the TOLM register.

5.7 IICH Register and Memory Mappings

This section covers the Intel® 3100 Chipset's IICH various address decoding ranges.

Warning: This section is for background purposes, and must not to be considered by implementers and validators as part of the behavioral definition of the Intel® 3100 Chipset. Each decode range is described elsewhere in the section associated with the corresponding function.

5.7.1 I/O Map

The I/O map is divided into separate types. Fixed ranges cannot be moved, but in some cases can be disabled. Variable ranges can be moved and can also be disabled.

5.7.1.1 Fixed I/O Address Ranges

Table 98 shows the Fixed I/O decode ranges from the CPU perspective. Note that for each I/O range, there may be separate behavior for reads and writes. Cycles that go to target ranges that are marked as Reserved will not be decoded, and are passed to PCI to PCI Bridge where they are dropped.

Address ranges that are not listed or marked reserved are NOT positively decoded by the IICH (unless assigned to one of the variable ranges). In subtractive mode, I/O ranges that are not otherwise decoded are forwarded to PCI to PCI Bridge where they are dropped.

Table 98. Fixed I/O Ranges Decoded by IICH (Sheet 1 of 3)

I/O Address	Read Target	Write Target	Internal Unit	Separate Enable/ Disable
00h – 08h	DMA Controller	DMA Controller	DMA	None
09h – 0Eh	RESERVED	DMA Controller	DMA	None
0Fh	DMA Controller	DMA Controller	DMA	None
10h – 18h	DMA Controller	DMA Controller	DMA	None
19h – 1Eh	RESERVED	DMA Controller	DMA	None
1Fh	DMA Controller	DMA Controller	DMA	None
20h – 21h	Interrupt Controller	Interrupt Controller	Interrupt	None
24h – 25h	Interrupt Controller	Interrupt Controller	Interrupt	None
28h – 29h	Interrupt Controller	Interrupt Controller	Interrupt	None
2Ch – 2Dh	Interrupt Controller	Interrupt Controller	Interrupt	None
2E – 2F	LPC SIO	LPC SIO	Forwarded to LPC	Yes
30h – 31h	Interrupt Controller	Interrupt Controller	Interrupt	None
34h – 35h	Interrupt Controller	Interrupt Controller	Interrupt	None
38h – 39h	Interrupt Controller	Interrupt Controller	Interrupt	None

Table 98. Fixed I/O Ranges Decoded by IICH (Sheet 2 of 3)

I/O Address	Read Target	Write Target	Internal Unit	Separate Enable/ Disable
3Ch – 3Dh	Interrupt Controller	Interrupt Controller	Interrupt	None
40h – 42h	Timer/Counter	Timer/Counter	PIT (8254)	None
43h	RESERVED	Timer/Counter	PIT	None
4E – 4F	LPC SIO	LPC SIO	Forwarded to LPC	Yes
50h – 52h	Timer/Counter	Timer/Counter	PIT	None
53h	RESERVED	Timer/Counter	PIT	None
60h	Microcontroller	Microcontroller	Forwarded to LPC	Yes w/ 64h
61h	NMI Controller	NMI Controller	Processor Interface	None
62h	Microcontroller	Microcontroller	Forwarded to LPC	Yes w/ 66h
63h	NMI Controller[1]	NMI Controller[1]	Processor Interface	Yes, alias to 61h
64h	Micocontroller	Microcontroller	Forwarded to LPC	Yes w/ 60h
65h	NMI Controller[1]	NMI Controller[1]	Processor Interface	Yes, alias to 61h
66h	Microcontroller	Microcontroller	Forwarded to LPC	Yes w/ 62h
67h	NMI Controller[1]	NMI Controller[1]	Processor Interface	Yes, alias to 61h
70h	RESERVED	NMI and RTC Controller	RTC	None
71h	RTC Controller	RTC Controller	RTC	None
72h	RTC Controller	NMI and RTC Controller	RTC	Yes, w/ 73h
73h	RTC Controller	RTC Controller	RTC	Yes, w/ 72h
74h	RTC Controller	NMI and RTC Controller	RTC	None
75h	RTC Controller	RTC Controller	RTC	None
76h	RTC Controller	NMI and RTC Controller	RTC	None
77h	RTC Controller	RTC Controller	RTC	None
80h	DMA Controller, or LPC, or PCI[2]	DMA Controller, or LPC, or PCI[2]	DMA	None
81h – 83h	DMA Controller	DMA Controller	DMA	None
84h – 86h	DMA Controller	DMA Controller and LPC or PCI[2]	DMA	None
87h	DMA Controller	DMA Controller	DMA	None
88h	DMA Controller	DMA Controller and LPC or PCI [2]	DMA	None
89h – 8Bh	DMA Controller	DMA Controller	DMA	None
8Ch – 8Eh	DMA Controller	DMA Controller and LPC or PCI [2]	DMA	None
8Fh	DMA Controller	DMA Controller	DMA	None
90h – 91h	DMA Controller	DMA Controller	DMA	Yes, alias to 8xh
92h	Reset Generator	Reset Generator	Processor Interface	None
93h – 9Fh	DMA Controller	DMA Controller	DMA	Yes, alias to 8xh
A0h – A1h	Interrupt Controller	Interrupt Controller	Interrupt	None
A4h – A5h	Interrupt Controller	Interrupt Controller	Interrupt	None
A8h – A9h	Interrupt Controller	Interrupt Controller	Interrupt	None
ACh – ADh	Interrupt Controller	Interrupt Controller	Interrupt	None
B0h – B1h	Interrupt Controller	Interrupt Controller	Interrupt	None
B2h – B3h	Power Management	Power Management	Power Management	None

Table 98. Fixed I/O Ranges Decoded by IICH (Sheet 3 of 3)

I/O Address	Read Target	Write Target	Internal Unit	Separate Enable/ Disable
B4h – B5h	Interrupt Controller	Interrupt Controller	Interrupt	None
B8h - B9h	Interrupt Controller	Interrupt Controller	Interrupt	None
BCh – BDh	Interrupt Controller	Interrupt Controller	Interrupt	None
C0h – D1h	DMA Controller	DMA Controller	DMA	None
D2h – DDh	RESERVED	DMA Controller	DMA	None
DEh – DFh	DMA Controller	DMA Controller	DMA	None
F0h	PCI and master abort	FERR#/IGNNE# / Interrupt Controller	Processor Interface	None
170h – 177h	SATA, or PCI[2]	SATA, or PCI[2]	SATA	Yes
1F0h – 1F7h	SATA, or PCI[2]	SATA, or PCI[2]	SATA	Yes
200 – 207h	Gameport Low	Gameport Low	Forwarded to LPC	Yes
208 – 20Fh	Gameport High	Gameport High	Forwarded to LPC	Yes
376h	SATA, or PCI[2]	SATA, or PCI[2]	SATA	Yes
3F6h	SATA, or PCI[2]	SATA, or PCI[2]	SATA	Yes
4D0h – 4D1h	Interrupt Controller	Interrupt Controller	Interrupt	None
CF9h	Reset Generator	Reset Generator	Processor Interface	None

Notes:
1. Only if the Port 61 Alias Enable bit (GCS.P61AE) bit is set. Otherwise, the target is PCI to PCI Bridge where it is dropped.
2. Transactions with PCI target are passed to internal PCI to PCI bridge where they are dropped

5.7.1.2 Variable I/O Decode Ranges

Table 99 shows the Variable I/O Decode Ranges. They are set using Base Address Registers (BARs) or other configuration bits in the various configuration spaces. The PnP software (PCI or ACPI) can use their configuration mechanisms to set and adjust these values.

Warning: The Variable I/O Ranges must not be set to conflict with the Fixed I/O Ranges. If the configuration software allows conflicts to occur, it may produce unpredictable results. There are no checks for conflicts.

Table 99. Variable I/O Decode Ranges (Sheet 1 of 2)

Range Name	Mappable	Size (Bytes)	Target
ACPI	Anywhere in 64 K I/O Space	64	Power Management
USB #1	Anywhere in 64 K I/O Space[1]	32	USB1 Host Controller 1
SMBus	Anywhere in 64K I/O Space	32	SMB Unit
TCO	96 bytes above ACPI base	32	TCO Unit
GPIO	Anywhere in 64 K I/O space	64	GPIO Unit
Parallel Port	3 ranges in 64 K I/O Space	8[2]	LPC Peripheral
Serial Port 1	8 Ranges in 64 K I/O Space	8	LPC Peripheral
Serial Port 2	8 Ranges in 64 K I/O Space	8	LPC Peripheral
Floppy Disk Controller	2 Ranges in 64 K I/O Space	8	LPC Peripheral
USB #2	Anywhere in 64 K I/O Space[1]	32	USB1 Host Controller 2

Table 99. Variable I/O Decode Ranges (Sheet 2 of 2)

Range Name	Mappable	Size (Bytes)	Target
USB #3	Anywhere in 64 K I/O Space[1]	32	USB1 Host Controller 2
USB #4	Anywhere in 64 K I/O Space[1]	32	USB1 Host Controller 2
LPC Generic 1	Anywhere in 64 K I/O Space	128	LPC Peripheral
LPC Generic 2	Anywhere in 64 K I/O Space	16	LPC Peripheral
I/O Trapping Ranges	Anywhere in 64 K I/O Space	1 to 256 Bytes	Trap on Internal I/O Data Bus

Notes:
1. These ranges are decoded directly. The I/O cycles will not be seen on PCI.
2. There is also an alias 400h above the parallel port range that is used for ECP parallel ports.

5.7.2 Memory Map

Table 100 shows (from the processor perspective) the memory ranges that are decoded. Cycles that arrive that are not directed to any of the internal memory targets that decode (see Table 100) are driven out on PCI to PCI Bridge where they are dropped.

Software must not attempt locks to the IICH's memory-mapped I/O ranges for USB 2.0, and HPET (High Precision Event Timer). If attempted, the lock is not honored which means potential deadlock conditions may occur.

Table 100. IICH Memory Decode Ranges (from CPU Perspective) (Sheet 1 of 2)

Memory Range	Target	Dependency/Comments
000E0000 - 000EFFFF	FWH	Bit 6 in FWH Decode Enable Register is set
000F0000 - 000FFFFF	FWH	Bit 7 in FWH Decode Enable Register is set
FEC00000 - FEC00040	I/O(x)APIC inside IICH	
FED40000 - FED40FFF	TPM on LPC	
FFC0 0000 - FFC7 FFFF FF80 0000 - FF87 FFFF	FWH (or PCI)[1]	Bit 8 in FWH Decode Enable Register
FFC8 0000 – FFCF FFFF FF88 0000 - FF8F FFFF	FWH (or PCI)[1]	Bit 9 in FWH Decode Enable Register
FFD0 0000 - FFD7 FFFF FF90 0000 - FF97 FFFF	FWH (or PCI)[1]	Bit 10 in FWH Decode Enable Register is set
FFD8 0000 – FFDF FFFF FF98 0000 - FF9F FFFF	FWH (or PCI)[1]	Bit 11 in FWH Decode Enable Register is set
FFE0 000 - FFE7 FFFF FFA0 0000 - FFA7 FFFF	FWH (or PCI)[1]	Bit 12 in FWH Decode Enable Register is set
FFE8 0000 – FFEF FFFF FFA8 0000 – FFAF FFFF	FWH (or PCI)[1]	Bit 13 in FWH Decode Enable Register is set
FFF0 0000 - FFF7 FFFF FFB0 0000 - FFB7 FFFF	FWH (or PCI)[1]	Bit 14 in FWH Decode Enable Register is set
FFF8 0000 – FFFF FFFF FFB8 0000 – FFBF FFFF	FWH (or PCI)[1]	Always enabled. The top two 64 kB blocks in this range can be swapped by the IICH. See Section 5.5 for details.
FF70 0000 - FF7F FFFF FF30 0000 - FF3F FFFF	FWH (or PCI)[1]	Bit 3 in FWH Decode Enable 2 Register is set
FF60 0000 - FF6F FFFF FF20 0000 - FF2F FFFF	FWH (or PCI)[1]	Bit 2 in FWH Decode Enable 2 Register is set

Table 100. IICH Memory Decode Ranges (from CPU Perspective) (Sheet 2 of 2)

Memory Range	Target	Dependency/Comments
FF50 0000 - FF5F FFFF FF10 0000 - FF1F FFFF	FWH (or PCI)[1]	Bit 1 in FWH Decode Enable 2 Register is set
FF40 0000 - FF4F FFFF FF00 0000 - FF0F FFFF	FWH (or PCI)[1]	Bit 0 in FWH Decode Enable 2 Register is set
1KB anywhere in 4GB range	USB 2.0 Host Controller	Enable via standard PCI mechanism (Device 29, Function 7)
FED0 X000h-FED0 X3FFh	HPET	BIOS determines "fixed" location which is one of four 1 KB ranges where X (in the first column) is 0h, 1h, 2h, or 3h.
All other	PCI	Any memory cycle in the first 4 GB page [bits 63:32 = 0s] will subtractively be forwarded to PCI to PCI bridge and dropped if not claimed by internal resources

Note:
1. Transactions that are targeted to PCI are dropped at PCI to PCI bridge.
2. PCI is the target when the Boot BIOS Destination selection bit is low (bit 3 of the General Control and Status Register). When PCI is selected, the FWH Decode Enable bits have no effect. Transactions that are targeted to PCI are dropped at PCI to PCI bridge.

5.7.3 Boot-Block Update Scheme

The IICH supports a "Top-Block Swap" mode, which causes the top memory block (the FWH boot block) to be swapped with another memory block. This allows for safe update of the Boot Block (even if a power failure occurs). When the "top-swap" enable bit is set, inverts A16 for cycles going to the upper two 64 Kbyte blocks in the FWH. Specifically, in this mode, accesses to FFFF_0000h-FFFF_FFFFh are directed to FFFE_0000h-FFFE_FFFFh and vice versa. When the Top Swap Enable bit is 0, the IICH will not invert A16. This bit is automatically set to 0 by RTEST#, but not by PLTRST#.

The scheme is based on the concept that the top block is reserved as the "boot" block, and the block immediately below the top block is reserved for doing boot-block updates.

The algorithm is:

1. Software copies the top block to the block immediately below the top.
2. Software checks that the copied block is correct. This could be done by performing a checksum calculation.
3. Software sets the "Top-Block Swap" bit. This will invert A16 for cycles going to the FWH.
4. Software erases the top block.
5. Software writes the new top block.
6. Software checks the new top block.
7. Software clears the top-block swap bit.
8. Software sets the Top_Swap Lock-Down bit.

If a power failure occurs at any point after step 3, the system is able to boot from the copy of the boot block that is stored in the block below the top. This is because the top-swap bit is backed in the RTC well.

Note: The top-block swap mode may be forced by an external strapping option (see Chapter 13.0, "IMCH Registers."). When top-block swap mode is forced in this manner,

the TOP_SWAP bit cannot be cleared by software. A reboot with the strap removed will be required to exit a forced top-block weap mode.

Note: Top-block swap mode only affects accesses to the Firmware Hub space, not feature space.

Note: The top-block swap mode has no effect on accesses below FFFE_0000h.

6.0 Platform Configuration

6.1 RASUM Features - SMBus and TAP Access

Configuration registers are accessible from either the host processor or from the SMBus. The processor will be able to access all configuration registers through host configuration cycles. Access via SMBus is read/write to the IMCH configuration registers. The SMBus cannot use the IMCH's SM-port target interface to access any register in the IICH or outside of the Intel® 3100 Chipset. Each device must have its own SMBus target port. The SMBus master implemented for PCI Express* Hot Plug is dedicated for the sole purpose of accessing the external I/O expander.

The Intel® 3100 Chipset does not shadow the RASUM registers for the SMBus. The PCI legacy registers associated with error reporting are not shadowed and are not sticky through reset. The PCI legacy registers are not cleared upon a read access. To clear these registers, a write access will need to be performed. The IMCH SMBus has full read/write access to the IMCH PCI legacy registers.

The IMCH global RASUM register set and those registers applicable to logical bus#0 and memory are implemented in Function 1 of Device 0. RASUM registers specific to other internal devices appear in the register map for the associated device. The IMCH error control registers are in Function 1, and are read/write accessible by the processor and through the SMBus. The IMCH error logging registers are also available to the processor and SMB master in Function 1. The IMCH RASUM control register and the “CMD” registers (SERRCMD, SMICMD, etc.), which control generation of SERR#, SMI#, and SCI#, are read/write accessible by the processor and through the SMBus.

FSB-initiated accesses to configuration space registers are serviced through configuration ring. It is perfectly legal for an SMBus access to be requested while a FSB-initiated access is already in progress. In other words, SMBus configuration accesses and processor configuration cycles may occur at the same time. The IMCH supports “wait your turn” arbitration to resolve all collisions and overlaps, such that the access that reaches the configuration ring arbiter first is serviced first while the conflicting access is held off. An absolute tie at the arbiter is resolved in favor of the FSB.

6.2 Platform Configuration Structure Conceptual Overview

The IMCH and IICH are physically connected by an internal interface called NSI (North South Interface). From a configuration standpoint, NSI is logically PCI bus #0. As a result, all devices internal to the IMCH and IICH, except host switch devices, appear to be on PCI bus #0. The system's primary PCI expansion bus is physically attached to the IICH and, from a configuration perspective, appears to be a hierarchical PCI bus behind a PCI-to-PCI bridge and therefore has a programmable Bus number. The PCI Express ports appear to system software to be real PCI busses behind PCI-to-PCI bridges that reside as devices on PCI bus #0.

The Intel® 3100 Chipset decodes multiple PCI Device numbers. The configuration registers for the devices are mapped as devices residing on PCI bus #0 except for host switch devices. Each Device Number may contain multiple functions. See Table 101, “PCI Devices and Functions on Bus 0” for device and function assignments.

Table 101. PCI Devices and Functions on Bus 0

Device	Function	Function Description
0	0	IMCH
0	1	IMCH, error status
0	2	Reserved
1	0	IMCH EDMA engine
2	0	IMCH PCI Express Port A0 (PEA) x8 or (PEA0) x4 unit
3	0	IMCH PCI Express Port A1 (PEA1) x4 unit
8	0	IMCH Test and Device 0 Overflow
9	0	Reserved
30	0	IICH PCI to PCI Bridge
31	0	IICH LPC Interface
31	2	IICH SATA Controller
31	3	IICH SMBus Controller
31	4	Reserved
31	5	Reserved
31	6	Reserved
29	0	IICH USB Controller 1
29	1	IICH USB Controller 2
29	7	IICH USB 2.0 Controller
28	0	IICH PCI Express Port B0
28	1	IICH PCI Express Port B1
28	2	IICH PCI Express Port B2
28	3	IICH PCI Express Port B3

6.2.1 IMCH PCI Devices

The PCI predefined header has five fields that deal with device identification. All devices are required to implement these fields. Generic configuration software is able to easily determine the device available for use. These registers are read only. The five fields are vendor ID, device ID, revision ID, header type, and class code:

- The 16-bit vendor ID is assigned by PCI SIG and has a value of 8086h for Intel.
- The 16-bit device ID is assigned by the vendor.
- The 8-bit revision ID is chosen by the vendor to indicate the different steppings of a device. The value 00h designates an A0 stepping.
- The header type specifies the structure of the second half of the header, and also whether or not the device has multiple functions. The value 80h indicates a multi-function device.
- The class-code field identifies the generic function of the device. The class-code is further broken into three sub-fields, base class, sub-class, and programming interface. The Intel® 3100 Chipset has a base class code of 06h indicating a bridge device. The sub-class value of 00h indicates a host bridge.

A disabled or non-existent IMCH device's configuration register space is hidden, returning all 1's for reads and dropping writes just as if the cycle terminated with a Master Abort on PCI.

If one or more IMCH devices or some of their functions are not supported on the platform, each can be disabled individually. When a device or function is disabled, it does not appear at all to the software (no responses to any register reads and no responses to any register writes). This is intended to prevent software from thinking that a device or function is present (and reporting it to the end-user).

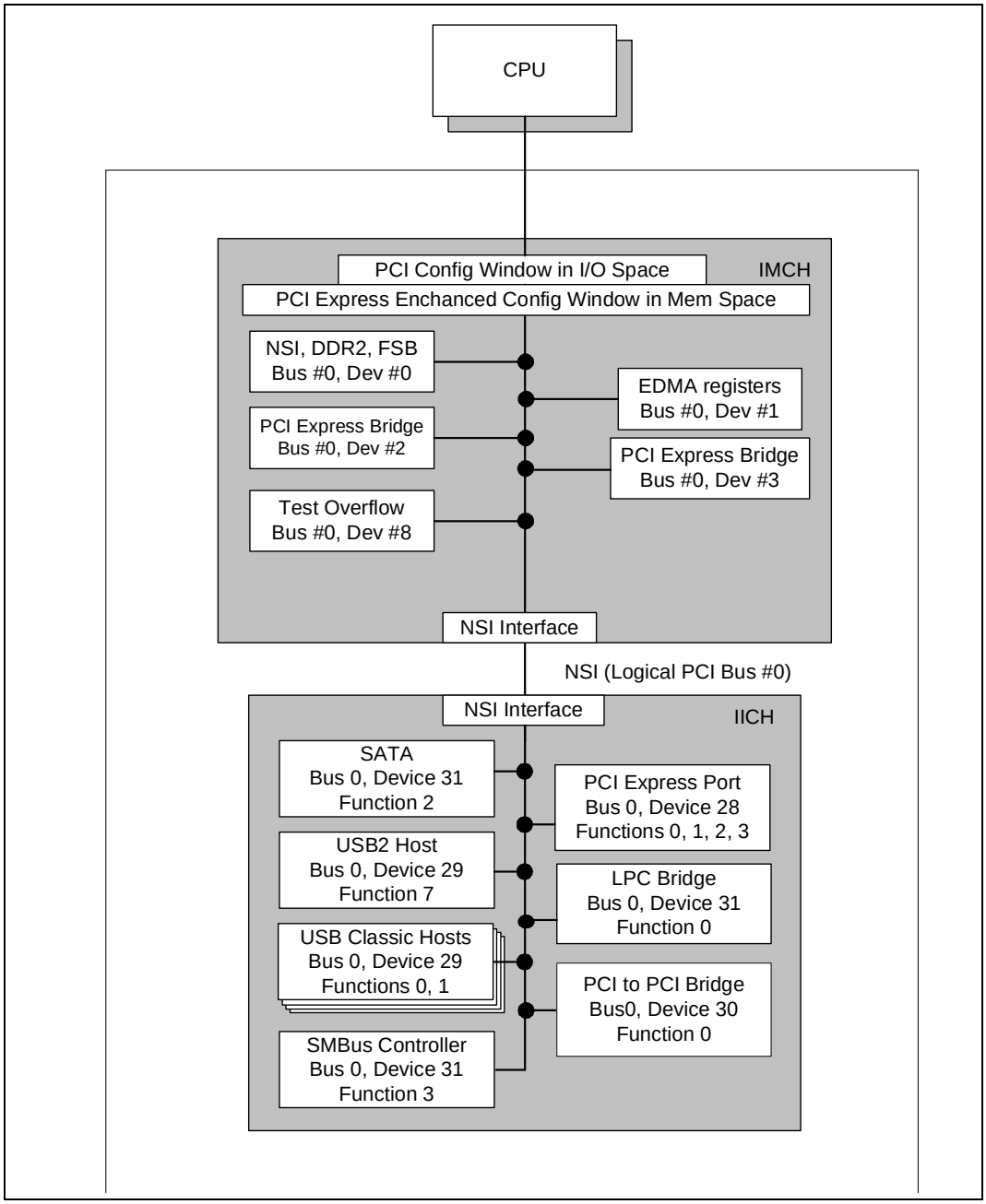
When a PCI Express interface is unpopulated or fails to train, the associated configuration register space is hidden. This returns all ones for all registers as if the cycle terminated with a Master Abort on PCI. Also, if PCI Express port PEA0 is configured for x8 operation rather than x4, the corresponding PCI Express port PEA1 configuration space will be hidden. In support of Hot-Plug capability, if the device is unpopulated or fails to train, the device could be removed and replaced with a working device, which precludes the removal of this configuration space for this possibility.

6.2.2 IICH PCI Devices

Logically, the IICH appears as multiple PCI devices within a single physical component also residing on PCI bus #0. One of the IICH devices is a PCI-to-PCI bridge. Logically, the primary side of the bridge resides on PCI #0 while the secondary is a standard PCI expansion bus.

Note: The internal devices in the IMCH and IICH (except host switch devices) logically constitute as PCI Bus #0 to configuration software (see Figure 44).

Figure 44. Bus 0 Device Map



6.3 Routing Configuration Accesses

The IMCH supports up to two x4 PCI Express interfaces:

- PEA0 and PEA1. These two interfaces can be combined to form a x8 interface, PEA.

The IMCH is responsible for routing PCI configuration cycles to the proper interface. PCI configuration cycles to IICH internal devices and downstream devices are routed to the IICH via the internal NSI bus. PCI configuration cycles to the IMCH PCI Express interfaces are routed to PEA(0:1). Routing of configuration accesses to PEA(0:1) is controlled via the standard PCI-to-PCI bridge mechanism using information contained within the PRIMARY BUS NUMBER, the SECONDARY BUS NUMBER, and the SUBORDINATE BUS NUMBER registers of the corresponding PCI-to-PCI bridge device.

A detailed description of the mechanism for translating processor I/O bus cycles to configuration cycles on one of the buses is described below.

Note: The IMCH supports a variety of connectivity options. When any of the IMCH's interfaces are disabled, the associated interface's device registers are hidden. All configuration cycles (reads and writes) to disabled devices on bus 0 are forwarded to the NSI where they will Master Abort.

6.3.1 Standard PCI Bus Configuration Mechanism

The PCI Bus defines a slot based "configuration space" that allows each device to contain up to eight functions with each function containing up to 256 8-bit configuration registers. The *PCI Specification* defines two bus cycles to access the PCI configuration space: Configuration Read and Configuration Write. Memory and I/O spaces are supported directly by the processor. Configuration space is supported by a mapping mechanism implemented within the IMCH. The *PCI Specification* defines two mechanisms to access configuration space, Mechanism 1 and Mechanism 2. Intel® 3100 Chipset supports Mechanism 1.

The configuration access mechanism makes use of the CONFIG_ADDRESS Register and CONFIG_DATA Register. To reference a configuration register a Dword (32-bit) I/O write cycle is used to place a value into CONFIG_ADDRESS that specifies the PCI bus, the device on that bus, the function within the device, and a specific configuration register of the device function being accessed. CONFIG_ADDRESS[31] must be a '1' to enable a configuration cycle. CONFIG_DATA then becomes a window into the four bytes of configuration space specified by the contents of CONFIG_ADDRESS. Any read or write to CONFIG_DATA will result in the IMCH translating the CONFIG_ADDRESS into the appropriate configuration cycle.

The IMCH is responsible for translating and routing the processor's I/O accesses to the CONFIG_ADDRESS and CONFIG_DATA registers to internal IMCH configuration registers, for NSI, and PCI Express ports PEA(0:1).

Note: It is only possible to generate 1-4 byte configuration accesses via this mechanism, which is in line with IMCH capabilities. The IMCH ***ONLY*** support accesses up to 1 Dword (32 bits) in size into the configuration register space (internal or external).

6.3.2 PCI Bus #0 Configuration Mechanism

The IMCH decodes the Bus Number (bits 23:16) and the Device Number fields of the CONFIG_ADDRESS register. If the Bus Number field of CONFIG_ADDRESS is 0, the configuration cycle is targeting a PCI Bus #0 device.

The Host-NSI Bridge entity within the IMCH is hardwired as Device #0 on PCI Bus #0.

The EDMA Controller within the IMCH is hardwired as Device #1 on PCI Bus #0.

The Host-PEA0 bridge entity within the IMCH is hardwired as Device #2 on PCI Bus #0.

The Host-PEA1 bridge entity within the IMCH is hardwired as Device #3 on PCI Bus #0.

Device #8 contains IMCH test configuration registers that would not fit in Device #0. Usually this space is not visible to the OS or applications.

Configuration cycles to any of the IMCH's enabled internal devices are confined to the IMCH and not sent over NSI. Accesses to disabled IMCH internal devices, or devices #10 to #31, are forwarded over NSI as Type 0 Configuration Cycles. A[1:0] of the NSI Request Packet for the Type 0 configuration cycle is "00". Bits 31:2 of the CONFIG_ADDRESS register is translated to the A[31:2] field of the NSI Request Packet of the configuration cycle as shown in Figure 45. The IICH decodes the Type 0 access and generates a configuration access to the selected internal device.

Figure 45. NSI Type 0 Configuration Address Translation

CONFIG_ADDRESS

31	24 23	16 15	11 10	8 7	2	1	0
1	Reserved	**0**	Device Number	Function	Register Number	x	x

NSI Type 0 Configuration Address Extension

31	28 27	24 23	16 15	11 10	8 7	2	1	0
Reserved			Device Number	Function	Register Number		0	**0**

6.3.3 Primary PCI and Downstream Configuration Mechanism

If the Bus Number in the CONFIG_ADDRESS is non-zero, and does not lie between the SECONDARY BUS NUMBER register and the SUBORDINATE BUS NUMBER register for one of the PCI Express ports or host switches, the IMCH will generate a Type 1 NSI Configuration Cycle. A[1:0] of the NSI request packet for the Type 1 configuration cycle is "01". Bits 31:2 of the CONFIG_ADDRESS register is translated to the A[31:2] field of the NSI request packet of the configuration cycle as shown in Figure 46. This NSI configuration cycle is sent over NSI.

If the cycle is forwarded to the IICH via NSI, the IICH compares the non-zero Bus Number with the SECONDARY BUS NUMBER and SUBORDINATE BUS NUMBER registers of its P2P bridges to determine if the configuration cycle is meant for the Primary PCI or one of the IICH's PCI Express ports.

Figure 46. NSI Type 1 Configuration Address Translation

CONFIG_ADDRESS

31	24 23	16 15	11 10	8 7	2	1	0
1	Reserved	**Bus Number**	Device Number	Function	Register Number	x	x

NSI Type 1 Configuration Address Extension

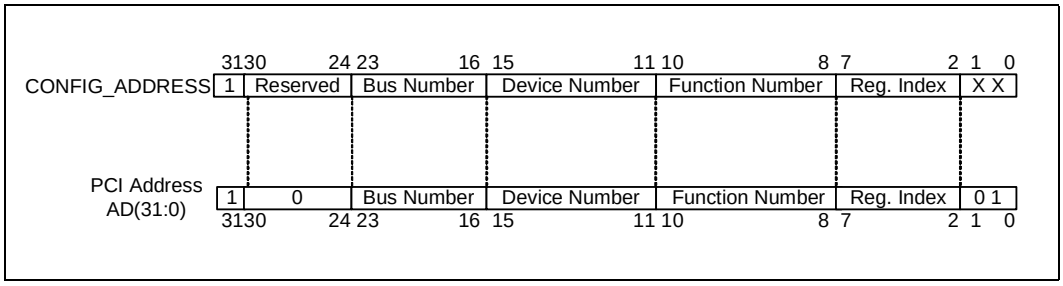
31	28 27	24 23	16 15	11 10	8 7	2	1	0
Reserved		**Bus Number**	Device Number	Function	Register Number		0	**1**

6.3.4 IMCH PCI Express Bus Configuration Mechanism

From Intel® 3100 Chipsetconfiguration perspective, the PCI Express ports are seen as PCI bus interfaces residing on a Secondary Bus side of the “virtual” PCI-to-PCI bridges referred to as the IMCH Host-PCI Express bridge. On the Primary bus side, the “virtual” PCI-to-PCI bridge is attached to PCI Bus #0. Therefore, the PRIMARY BUS NUMBER register is hardwired to “0”. The “virtual” PCI-PCI bridge entity converts Type #1 PCI Bus Configuration cycles on PCI Bus #0 into Type 0 or Type 1 configuration cycles on the PCI Express interfaces. Type 1 configuration cycles on PCI Bus #0 that have a BUS NUMBER that matches the SECONDARY BUS NUMBER of one of the IMCH’s “virtual” P2P bridges are translated into Type 0 configuration cycles on the appropriate PCI Express interface. The address bits are mapped as described in Figure 47.

If the Bus Number is non-zero, greater than the value programmed into the SECONDARY BUS NUMBER register, and less than or equal to the value programmed into the corresponding SUBORDINATE BUS NUMBER register the configuration cycle is targeting a PCI bus downstream of the targeted PCI Express interface. The IMCH will generate a Type 1 configuration cycle on the appropriate PCI Express interface. The address bits are mapped as described in Figure 47.

Figure 47. Mechanism 1 Type 1 Configuration Address to PCI Address Mapping



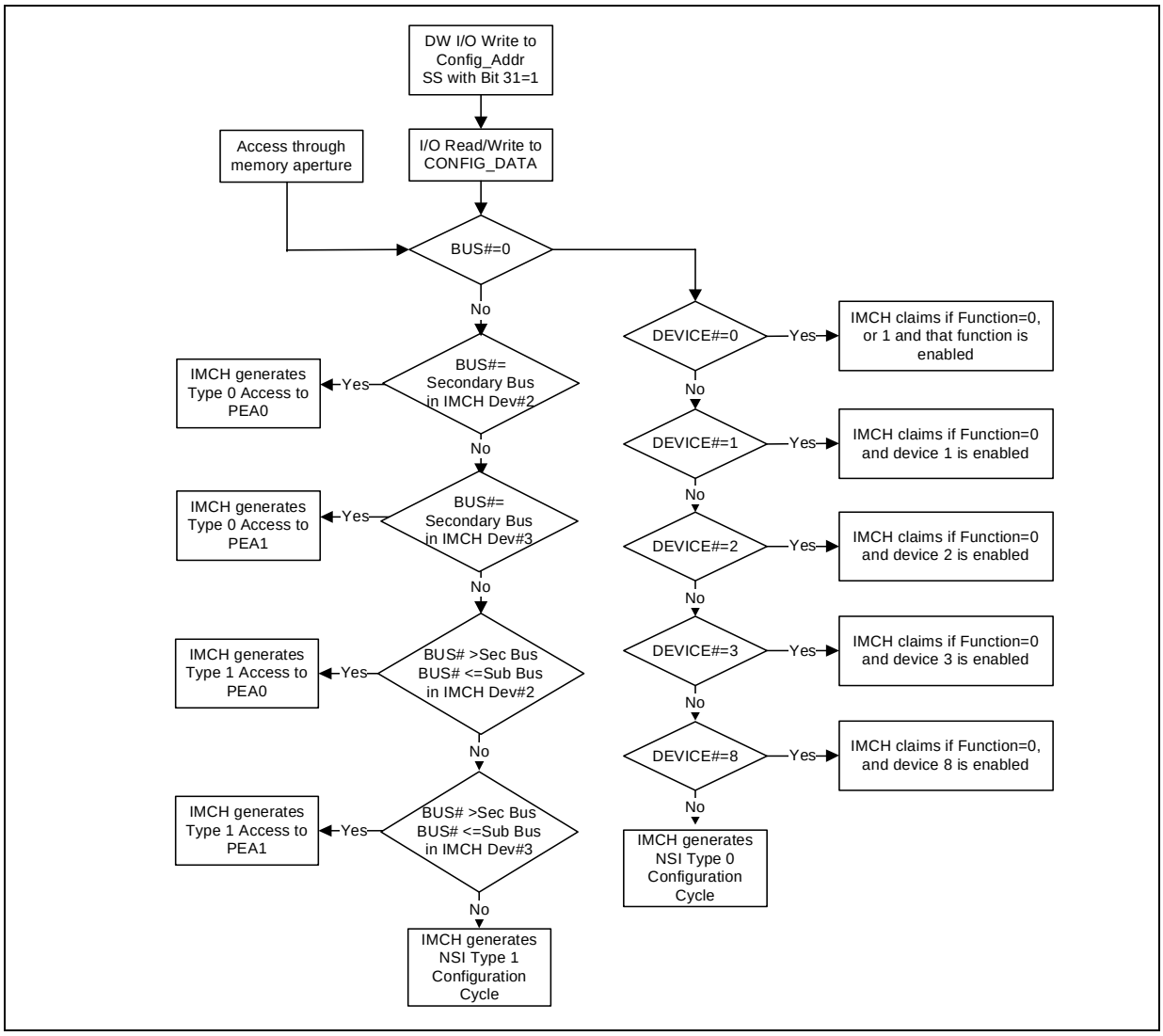
To prepare for mapping of the configuration cycles on PCI Express the initialization software will go through the following sequence:

Scan all devices residing on the PCI Bus #0 using Type 0 configuration accesses.

For every device residing at bus #0 which implements PCI-to-PCI bridge functionality, it will configure the secondary bus of the bridge with the appropriate number and scan further down the hierarchy. This process will include the configuration of the “virtual” PCI-to-PCI bridges within the IMCH used to map the PCI Express device’s address spaces in a software specific manner.

6.3.5 IMCH Configuration Cycle Flow Chart

Figure 48. IMCH Configuration Flow Chart



6.4 IMCH Register Introduction

The IMCH contains two sets of software accessible registers, accessed via the Host processor I/O address space: control registers I/O mapped into the processor I/O space, which control access to PCI configuration space, and internal configuration registers residing within the IMCH, which are partitioned into multiple logical device register sets (“logical” since they reside within a single physical device).

The IMCH internal registers (I/O Mapped and Configuration registers) are accessible by the Host processor. The registers can be accessed as Byte, Word (16-bit), or Dword (32-bit) quantities, with the exception of CONFIG_ADDRESS, which can only be accessed as a Dword. All multi-byte numeric fields use “little-endian” ordering (i.e., lower addresses contain the least significant parts of the field).

Note: Irrespective of the access mechanism used (I/O register mechanism, or memory-mapped mechanism), the IMCH ***ONLY*** supports 1-4 byte accesses into configuration space. Software must (if necessary) take steps to prevent use of opcodes that would treat configuration space destinations as objects greater than a single Dword (32 bits) in size. Such attempted usage will result in spurious behavior up to and including hanging the platform.

Some of the IMCH registers described in this section contain reserved bits which are labeled "Reserved". Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions and then written back. Note the software does not need to perform read, merge, write operation for the configuration address register.

In addition to reserved bits within a register, the IMCH contains address locations in the configuration space of the Host-NSI Bridge entity that are marked either "Reserved" or "Intel Reserved". The IMCH responds to accesses to "Reserved" address locations by completing the host cycle. When a "Reserved" register location is read, a zero value is returned. ("Reserved" registers can be 8-, 16-, or 32-bit in size). Write operations to "Reserved" registers have no effect on the IMCH. Registers that are marked as "Intel Reserved" must not be modified by system software. Writes to "Intel Reserved" registers may cause system failure. Reads to "Intel Reserved" registers may return a non-zero value.

After a reset, the Intel® 3100 Chipset sets its entire internal configuration registers to predetermined default states. At reset, some register values are determined by external strapping options. A register's default value represents the minimum functionality feature set required to successfully bring up the system. It is the responsibility of the system initialization software (usually the BIOS) to properly determine the DRAM configurations, operating parameters and optional system features that are applicable, and to program Intel® 3100 Chipset registers accordingly.

6.5 IMCH Sticky Registers

Certain registers in the IMCH are sticky through a hard-reset. They will only be reset on a Power-good reset. In general, these registers are the error logging registers and a few special cases. The error command registers are not sticky, so that on reset bogus errors are not reported and that errors are not reported through a mechanism that hasn't been set up in code yet. Only those registers that are explicitly marked as "Sticky: YES" are sticky. Those not marked or those marked as NO are not sticky.

The following registers are sticky:

- Device 0, Function 0: Critical DRAM control registers, including DRM, a portion of DRC, DRT clock gearing and clock disable registers
- Device 0, Function 0: ECO sticky register
- Device 0, Function 0, Bar 14: BIOS notepad sticky register
- Device 0, Function 1: error information registers (not the command registers)
- Device 0, Function 2: error status and transmit driver enable bits
- Device 2, Function 0: error information registers (not the command registers)
- Device 3, Function 0: error information registers (not the command registers)
- Device 8 Function 0: power on configuration bits

6.6 IMCH I/O Mapped Registers

The IMCH contains two registers that reside in the processor I/O address space – the Configuration Address (CONFIG_ADDRESS) Register and the Configuration Data (CONFIG_DATA) Register. The Configuration Address Register enables/disables the configuration space and determines what portion of configuration space is visible through the Configuration Data window.

6.6.1 CONFIG_ADDRESS - Configuration Address Register

CONFIG_ADDRESS is a 32-bit register that can be accessed only as a Dword. A Byte or Word reference will "pass through" the Configuration Address Register and NSI onto the IICH as an I/O cycle. The CONFIG_ADDRESS register contains the Bus Number, Device Number, Function Number, and Register Number for which a subsequent configuration access is intended.

Table 102. CONFIG_ADDRESS - Configuration Address Register

Offset	0CF8h Accessed as a Dword	***Attribute:***	Read/Write
Default Value:	00000000h	***Size:***	32 bit

Bits	Name	Description	Reset Value	Access
31	CFGE	**Configuration Enable.** 0 = Accesses to PCI configuration space are disabled. 1 = Accesses to PCI configuration space are enabled.	0	RW
30:24	Reserved	**Reserved.** These bits are read only and have a value of 0.	0	
23:16	Bus Number	Contains the bus number being targeted by the configuration cycle.	0	RW
15:11	Device Number	Selects one of the 32 possible devices per bus.	0	RW
10:08	Function Number	Selects one of eight possible functions within a device.	0	RW
07:02	Register Number	This field selects one register within the particular Bus, Device, and Function as specified by the other fields in the Configuration Address Register. This field is mapped to A[07:02] during NSI or PCI Express Configuration cycles.	0	RW
01:00	Reserved	**Reserved**	0	

6.6.2 CONFIG_DATA - Configuration Data Register

CONFIG_DATA is a 32-bit read/write window into configuration space. The portion of configuration space that is referenced by CONFIG_DATA is determined by the contents of CONFIG_ADDRESS.

Table 103. CONFIG_DATA Configuration Data Register

Offset	0CFCh	***Attribute:***	Read/Write
Default Value:	00000000h	***Size:***	32 bits

Bits	Name	Description	Reset Value	Access
31:00	CDW	**Configuration Data Window.** If bit 31 of CONFIG_ADDRESS is one, any I/O access that to the CONFIG_DATA register is mapped to configuration space using the contents of CONFIG_ADDRESS.	0	RW

6.7 IMCH Memory Mapped Registers

Certain DRAM compensation control, EDMA control/status registers, NSI control/status and PCI Express will reside in memory mapped space instead of configuration space. These memory mapped address regions are setup through base address registers and capability pointers, which will reside in configuration address space. These registers are documented in the configuration register chapter. These base address registers follow the standard definition as found in the *PCI Express Specification*.

These memory mapped register regions must not be marked as WC (Write-Combining), as all accesses to the registers within these regions are limited to Dword access, and write-combining is not allowed. Further, these registers must not be accessed utilizing processor operations with a data operand size greater than 32-bits, as such access is strictly unsupported by the IMCH.

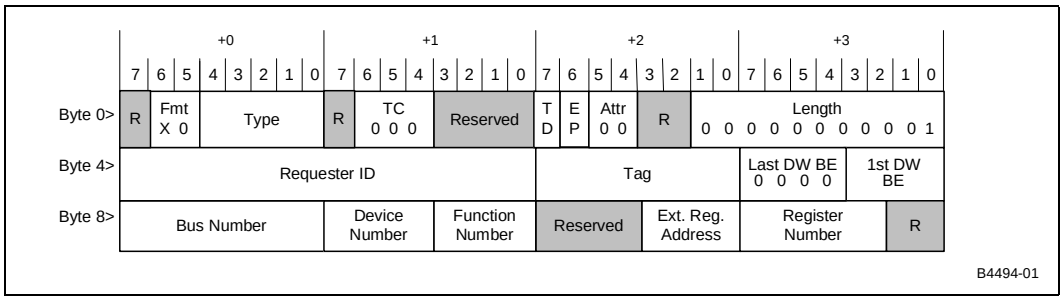
6.8 PCI Express* Enhanced Configuration Mechanisms

PCI Express extends the configuration space to 4096 bytes per device/function as compared to 256 bytes allowed by PCI 2.2 configuration space. PCI Express configuration space is divided into a PCI 2.2 compatible region, which consists of the first 256 B of a logical device's configuration space and an extended PCI Express region which consists of the remaining configuration space. The PCI 2.2 compatible region can be accessed using either the mechanisms defined in the *PCI Specification* Rev. 2.2 or using the enhanced PCI Express configuration access mechanism. All changes made using either access mechanism are equivalent; however, software is not allowed to interleave PCI Express and PCI access mechanisms to access the configuration registers of devices. The extended PCI Express region can only be accessed using the enhanced PCI Express configuration access mechanism.

6.8.1 PCI Express* Configuration Transaction Header

The PCI Express Configuration Transaction Header includes an additional four bits for the Register Number field (ExtendedRegisterAddress[3:0]) to provide additional configuration space.

Figure 49. PCI Express* Configuration Transaction Header



The PCI 2.2 compatible configuration access mechanism uses the same Request format as the enhanced PCI Express mechanism. For PCI compatible Configuration Requests, the Extended Register Address field must be all zeros.

To maintain compatibility with PCI configuration addressing mechanisms, system software must access the enhanced configuration space using Dword operations (Dword-aligned) only.

6.8.2 Enhanced Configuration Hardware Implications

The IMCH must translate the memory-mapped extended enhanced PCI Express configuration access cycles from the host processor to PCI Express configuration cycles.

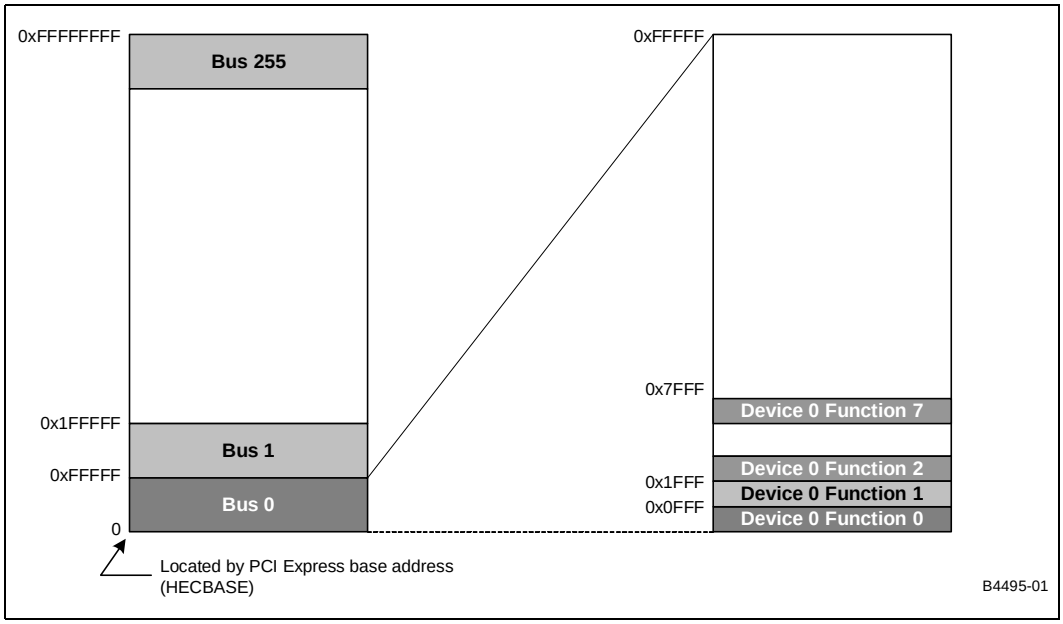
Devices are required to respond to an additional four bits for decoding configuration register access. Devices must decode the ExtendedRegisterAddress[3:0] field of the Configuration Request Header. This field is used in conjunction with the Register Number to specify the Dword address of the register being accessed.

A PCI Express device must be able to operate with basic required functionality in a legacy environment without requiring access to any extended PCI Express configuration.

6.8.3 Enhanced Configuration Memory Address Map

The Enhanced Configuration Memory Address Map is positioned into Intel® 3100 Chipset memory space by use of the PCI Express Enhanced Configuration Base register known as HECBASE. This register contains the address that corresponds to bits 31 to 28 of the base address for PCI Express enhanced configuration space below 4 GB. Configuration software will read this register to determine where the 256 Mbyte range of memory addresses resides for enhanced configuration. This register defaults to a value of E, which corresponds to E000 0000 for the IMCH. It is not intended that this value is ever changed by BIOS.

Figure 50. Enhanced Configuration Memory Address Map



6.8.4 Enhanced Configuration FSB Address Format

Table 104 presents the enhanced configuration address format for the front side bus. Note that bits 31:28 of Table 104 correspond to the default value of HECBASE.

Table 104. Enhanced Configuration FSB Address Format

Bits	Description
35:32	0h
31:28	Eh
27:20	Bus Number
19:15	Device Number
14:12	Function Number
11:00	Register Offset

7.0 RAS Features and Exception Handling

The Intel® 3100 Chipset is designed to bring enterprise level Reliability, Availability, Serviceability, Usability, and Manageability (RASUM) to the Embedded platform.

7.1 RAS Features

7.1.1 Data Protection

Due to the nature of having various data protection schemes on the different interfaces (ECC, parity, and CRC) it is necessary to be able to convert between them when transferring data internally. To accomplish this, protection of internal data is done with parity.

7.1.1.1 DRAM ECC

The DRAM interface uses a standard SEC/DED ECC across a 64-bit data quantity.

7.1.1.2 FSB Interface Signals

Address/request, response, and data bus signals are protected by parity. The address/ request and data busses can be configured to perform no error checking.

The FSB data parity scheme is not straightforward parity, so a description is warranted. Table 105 is slightly different from the processor specification to provide better readability. The data in a given clock cycle is quad-pumped while the parity (DP[3:0]#) that corresponds to this same data is common-clocked driven in the clock following the presentation of the data. The four sub-phases correspond to the data and data inversion bits that are driven out during a single clock cycle. The corresponding parity bits are calculated by XORing the four table components. For example, DP3# is driven out in the clock following the data is an XOR of DP3a, DP3b, DP3c, and DP3d. Table 105 shows these four parity components in different diagonals of the table. Ultimately, an approximate 32 bytes of data is protected by four parity bits. This particular rotating parity scheme is able to detect stuck at faults more readily than standard parity schemes.

Table 105. FSB Matrix Parity Scheme

Data Signals	Sub phase 1	Sub phase 2	Sub phase 3	Sub phase 4
D[15:0]#, DINV0#	DP3a	DP2b	DP1c	DP0d
D[31:16]#, DINV1#	DP0a	DP3b	DP2c	DP1d
D[47:32]#, DINV2#	DP1a	DP0b	DP3c	DP2d
D[63:48]#, DINV3#	DP2a	DP1b	DP0c	DP3d

7.1.1.3 PCI Express* Interface

These high-speed serial interfaces have traditional CRC protection. The data packets utilize a 32-bit CRC protection scheme, specifically the same CRC-32 used by Ethernet - 0x04C11DB7. The smaller and less error-prone link packets utilize a 16-bit CRC scheme. Since packets utilize 8B/10B encoding and not all encodings are used, this provides further data protection because illegal codes can be detected. Also, if errors are detected on the reception of data packets due to various transients, these data packets can be retransmitted. Hardware logic supports this link-level retry without software intervention.

PEA has the additional functionality of ECRC. ECRC is used for end-to-end protection except when a packet is switched from one PCI Express (PEA) port to another peer-to-peer posted write. The Intel® 3100 Chipset is always seen as an end point for ECRC. For example, inbound Intel® 3100 Chipset is the destination for ECRC and outbound Intel® 3100 Chipset is the source of ECRC.

7.1.1.4 Data Error Propagation Between Interfaces/Units

Due to the nature of having various data protection schemes; ECC, parity, and CRC - it is necessary to be able to convert between the separate schemes. Beyond this requirement, it is necessary to indicate whether or not incoming data is corrupted. Also, it is useful to know when internal data has been corrupted during transit. To accomplish this, the IMCH uses parity to protect internal data. This requires units to add two parity bits for each 64 bits of data path width. Data received by a unit from outside the chip creates two parity bits to travel with the data, one provides parity on the upper 32 bits, and the other provides parity on the lower 32 bits. If either of the 32-bit halves is required to be poisoned, both halves are poisoned. This provides the user of the data a mechanism to recognize when a bit was flipped in transit by detecting when only one of the parity bits is corrupt. The user will flag this error condition as well as mark both halves bad. This covers both cases of the data starting out as either good or bad. If it started out as good, but a bit was flipped, it is indeed corrupted and must be marked as such. In the case where it started out as bad, and a bit was flipped, it is still corrupted, although probably a different data value than its starting value. This scheme works when all quantities being passed are 64 bits or greater. If a data path must be padded, it must be padded with 0's. Even parity will be used for this scheme; meaning that the total number of asserted bits including the parity bits is an even number of bits. This parity protection scheme applies to different interfaces on the chip hence the name: "Chip Two Bit Parity" or CTB parity.

Note: Due to EDMA byte realignment and parity manipulation, a single CTB parity bit error observed by the EDMA unit may poison either 2 or 4 DWords depending on the resultant alignment. Refer to the EDMA chapter for more details.

7.1.2 DRAM Data Integrity

7.1.2.1 Periodic Memory Scrubbing

A special DRAM memory scrubbing unit will walk through all DRAM, on a periodic basis, doing reads. Correctable errors found by the read are corrected and then the good data written back to DRAM. A write is only performed when a single bit error has been detected and is correctable, except when an incoming write to the same memory address is detected. In this case the scrub write is dropped and the scrub counter is advanced since this location is already being written. These transactions are treated as non-coherent, since these addresses are not placed on the FSB. The scrub unit starts at address 0 when initially enabled. Every 32 k clocks the unit will scrub one line and then increment the address by 64B or one cache line. Using this method, a 16 Gbyte system can be completely scrubbed in less than a day. (The cumulative effect of these scrub

writes do not cause any noticeable degradation to memory bandwidth, although they will cause a greater latency for that one very infrequent read that is delayed due to the scrub write cycle.)

7.1.2.2 DRAM Hardware Initialization

Hardware will be used to initialize main memory under the direction of BIOS. Once BIOS has programmed the IMCH with the DIMM profile, and has configured and calibrated the IMCH and populated DIMMs, it can utilize the DCALCSR register interface (see Section 13.7.1.3, "Offset 100 - 103h: DCALCSR – DCAL Control and Status Register") to initialize and/or test populated memory. The initialization of the DCALCSR will traverse the target range of memory addresses as rapidly as possible, providing an order of magnitude performance improvement over CPU-generated initialization or test.

The DCAL engine can be configured to choose values other than zero. The eight fixed hex data values selectable are alternating pairs of 0/F, A/5, 3/C, or 6/9. Alternate modes are provided in which LFSR random data may be used, or software explicitly specifies the full pattern of bits to be written in a collection of DCALDATA registers (see Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0") with or without a bit-shift left after every write. In all cases of pattern based initialization and test, the DCAL function does NOT calculate ECC on the fixed pattern or programmed value to be written across the target address range. Rather, the fixed pattern is extended to cover the data devices as well as the ECC devices in the target DIMM, and a strict bit-wise comparison is utilized to determine whether read-back verification passes or fails.

Once all desired testing has been completed, requirements dictate that memory be completely initialized to "0" prior to transferring control to the operating system. To accomplish this, BIOS must clear all the DCALDATA registers and utilize the explicit pattern mode of DCAL; such that each operational channel receives a pattern with 64 bits of "0" data, and eight bits of "0" ECC. It is possible to initialize memory a rank at a time, or en-masse, at the discretion of BIOS.

7.1.2.3 Uncorrectable Retries

If correctable errors are detected, the errors are in fact corrected and good data is presented to the requestor. If however, uncorrectable errors are detected, the request to memory will be repeated in order to cover the case where a transient caused the temporary error, and the subsequent read will deliver good data. Retries for a particular address will only occur once, and if the uncorrectable error is repeated it will be logged and escalated as directed by device configuration.

7.1.2.4 DRAM Refresh

As with any DRAM device, the storage element is inherently leaky, and must be recharged periodically to avoid loss of data integrity. Circuitry in the memory subsystem will ensure that refresh cycles occur in a periodic fashion across all active DIMMS to meet the specific DRAM requirements.

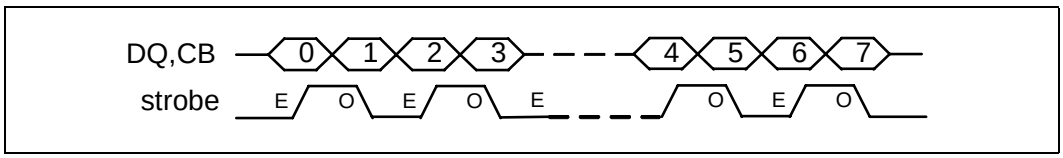
7.1.2.5 DCAL Debug Functionality

Another feature of the DRAM calibration logic is that it can be configured to perform an error monitor function. This error monitor captures both the failing and retried data of a DED retry event.

Terminology definitions are described to aid in the discussions and tables that follow. DQ refers to the data, and CB refers to the ECC check bits using DDR2 terminology. To follow the terminology used in the tables, keep in mind that a cacheline is delivered with 2 sets of four bursts of data. These four bursts are sometimes referred to as

bursts 0, 1, 2, and 3. But because we want to limit our discussion to a half cacheline which could be either burst 0 and 1 or 2 and 3, we will refer to the data with respect to the rising or falling edge of a representative single strobe. The rising edge of this strobe captures even data, which could be either Qwords 0 and 2 or Qwords 4 and 6. The falling edge of this strobe captures the odd data, which could be either Qwords1 and 3 or Qwords 5 and 7. Figure 51 illustrates this behavior.

Figure 51. DDR2 Data Bursts



7.1.3 PCI Express* Data Integrity

The PCI Express interfaces will incorporate several features to make this interface as robust as possible without software intervention.

7.1.3.1 PCI Express*Training

To establish a connection between PCI Express endpoints, they both participate in a sequence of steps known as training. This sequence will establish the operational width of the link as well as adjust skews of the various lanes within a link so that the data sample points can correctly take a data sample off of the link. The x4 link pairs capable of collapsing to x8 will first attempt to train independently, and will collapse to a single link at the x8 width upon detection of a single device returning link ID information upstream. Once the number of links has been established, they will negotiate to train at the highest common width, and will step down in its supported link widths in order to succeed in training. The ultimate result may be that the link has trained as a x1 link. Although the bandwidth of this link size is substantially lower than a x8 link or even a x4 link, it will allow communication between the two devices. Software will then be able to interrogate the device at the other end of the link to determine why it failed to train at a higher width, something that would not be possible without support for the x1 link width. It should be noted that width negotiation is only done during training or retraining, but not recovery.

7.1.3.2 PCI Express* Retry

The PCI Express interface incorporates a link level retry mechanism. The hardware detects when a transmission packet is corrupted and a retry of that particular packet and all following packets will be performed. Although this will cause a temporary interruption in the delivery of packets, it does so in order to maintain the link integrity.

7.1.3.3 PCI Express* Recovery

When numerous errors occur, the hardware may determine that the quality of the connection is in question, and the end points can enter a quick training sequence known as recovery. The width of the connection will not be renegotiated, but the adjustment of skew between lanes of the link may occur. This occurs without any software intervention, but the software may be notified.

7.1.3.4 PCI Express* Retrain

If the hardware is unable to perform a successful recovery then the link will automatically revert to the polling state, and initiate a full retraining sequence. This is a drastic event with an implicit reset to the downstream device and all subordinate devices, and is logged by the Intel® 3100 Chipset as a “Link Down” error. If escalation

of this event is enabled, software is notified of the link DL_DOWN condition. Once software has to be involved, then data will likely be lost, and processes need to be restarted, but this is still preferred to having to shut the system down, or go offline for an extended period of time.

7.1.4 Test/Support Major Busses

7.1.4.1 IMCH TAP

The IMCH supports full access to internal configuration registers via a Test Access Port (TAP). Systems and test fixtures so enabled will be able to make use of this bus. Accesses to devices marked as not present are still possible through the IMCH TAP.

7.1.4.2 IICH XOR

The IICH supports XOR Chain test mode. This non-functional test mode is a dedicated test mode when the chip is not operating in its normal manner.

7.1.4.3 SMB (IMCH)

The IMCH SMB is SMBus 2.0 compliant and it is compatible with most 2-wire components that are also I^2C compatible. Full access to internal configuration registers via the System Management Bus is supported. This will allow a server management card to control system configuration and to read various error/status information. Accesses to devices marked as not present will still be possible through SMB.

7.1.4.4 SMB (IICH)

The IICH SMB is SMBus 2.0 compliant and it is compatible with most 2-wire components that are also I^2C compatible. This SMBus interface can function as a host or target. This IICH SMB does not support access to internal configuration registers.

7.1.4.5 I^2C

Access to the external DIMMs will be through the IICH, via I^2C. This will be used to determine the nature of the DIMMs present in order to configure the memory subsystem correctly.

7.2 Exception Handling

There are a variety of exception conditions. Some are internally detected; some are detected on input pins; some are passed on behalf of other devices. All recognized exceptions eventually cause the IMCH to do one of the following: Send a SERR message, send a SCI message , send a SMI message, assert MCERR# on the front side bus, or do nothing. There is no determination of which errors go to which of the three error message schemes; it merely provides the capability for all combinations. It is the responsibility of the BIOS to determine the ultimate error reporting scheme. There will be an attempt to classify errors to whether they are fatal or non-fatal to more closely match the enterprise error presentation.

7.2.1 FERR/NERR Global Register Scheme

Figure 52. Global FERR/NERR Register Representation

Fatal (14b)	Non-Fatal (14b)	Reserved (4b)

The Global FERR register consists of three fields. The first or fatal field has 14b indicates the first signaled fatal global error from 14 different units. The second or non-fatal field indicates the first non-fatal global error that occurs from the same 14 different units. A non-fatal error may be either correctable or uncorrectable, but not fatal. These two fields usually have at most one bit asserted in each field. In the event of simultaneous errors occurring in the same core clock, more than one bit in a field may be set. The third 4-bit field is reserved for future enhancements.

The Global NERR register consists of these same three fields with slightly different functionality. Instead of just the first fatal or non-fatal global errors recorded, this register indicates the second, third, fourth, etc. global errors that are reported by the IMCH.

These two registers do not indicate what the error was, just indicates the severity of the error and what unit has more specific error information. Refer to Chapter 2.0, "Configuration Register Descriptions," for bit definitions in this register.

7.2.1.1 FERR/NERR Unit Registers

Each major unit will have a minimum of a pair of registers, known as the first error (FERR) and next error (NERR). Each unit has different and specific error bit definitions, and provides the specific type of error; information that is not found in the global registers. It is important to note that the unit FERR/NERR registers are simpler than the global for purposes of reuse and ease of implementation. While the global FERR register has a fatal and a non-fatal field, which lock down separately, the unit FERR register only has one field. The unit is however still required to send out separate fatal and non-fatal indications to the global FERR register if they detect both classifications of errors. Some units will support only one type. A unit that doesn't detect errors would not support either type.

7.2.1.2 Clearing FERR/NERR Registers

The following write-up is the recommended guideline to minimize the loss of error information.

For a given FERR/NERR register pair, first the FERR is read and cleared, then the NERR is read and cleared. This sequence is true for either the global FERR/NERR register pair or any given unit. Any errors occurring after the FERR is cleared will then cause the FERR to have a non-zero value.

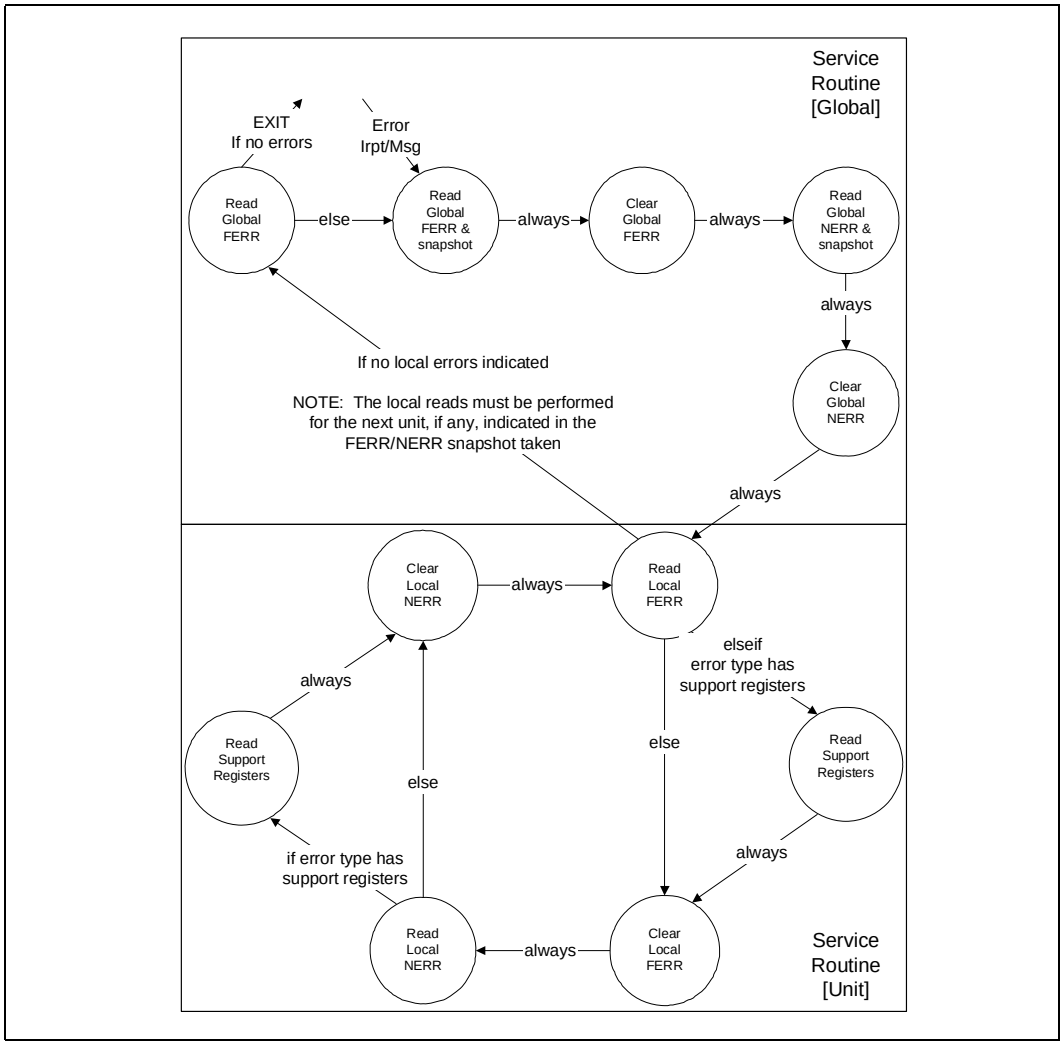
After the global FERR/NERR register pair is cleared, the unit FERR/NERR register pairs are interrogated, but only those indicated by the global FERR/NERR registers. Once the unit pair has been cleared, the unit FERR can be read again to ensure that no errors occurred during this local unit sequence. After the first unit FERR/NERR register pair has been serviced, this same sequence is performed for all other unit FERR/NERR register pairs that indicated errors in the global FERR/NERR registers. Once all unit error registers have been serviced, the final step is to read the global FERR register to determine if all system errors have been serviced. It is possible that errors could have occurred for a particular unit after that unit was serviced during the error routine, or that a unit had errors after the reading of the global FERR/NERR registers.

This is a simplified guideline, as particular errors will require additional registers to be read to gain complete information. Also, the PEA units have register hierarchy below the FERR/NERR registers that are discussed in a later section of this chapter.

When clearing errors, software must clear all the FERR/NERR bits in the local interface registers before clearing the global FERR/NERR registers. If the local registers are not cleared first, then the global FERR/NERR registers will latch the same error again as soon as they are cleared. This implementation allows software to clear the local FERR/

NERR registers, and then go clear the global FERR/NERR. S/W then reads back the global FERR/NERR and if it is non-zero, then a new error has occurred. If the global FERR/NERR has no bits set, then there are no more system errors.

Figure 53. FERR/NERR Service Routine



7.2.1.3 FERR/NERR Unit Specific

Each unit has different and specific error bit definitions, which are explained in Chapter 2.0, “Configuration Register Descriptions”.

7.2.1.4 SERR/SMI/SCI Enabling Registers

Each error reported has a full matrix of direction as to what error message it generates. For each unit FERR/NERR pair there are three more registers that enable each error for one of the three specific error messages. The logic does not appear to preclude the generation of all three messages for a single error, but this would not be a recommended configuration, and this needs to be looked into further. SERR stands for system error and is for reporting address and data parity errors, or any other

catastrophic system error. SCI stands for system control interrupt and is a shareable interrupt used to notify the OS of ACPI events. SMI stands for System Management Interrupt and is an OS-transparent interrupt generated by events on legacy systems.

7.2.1.5 MCERR Enabling Registers

An additional entry to the matrix of error signaling paths is the MCERR (machine check error) enabling register. In addition to the SERR, SMI, and SCI enabling registers, the MCERR enabling register allows the occurrence of an error to result in the MCERR# signal to be asserted on the front side bus. Machine check error is asserted to indicate an unrecoverable error, which is not a bus protocol violation.

7.2.1.6 Error Escalation Register

Since all error bits in the error registers are fully configurable, meaning that a given error can be configured to go to any of the four messaging methods, no global error escalation mechanism is required. Although, the errors occurrence is accumulated in the global FERR/NERR registers, all error messaging is initiated from the units themselves, and not from a central location.

7.2.1.7 Error Masking

A new feature being added for the Intel® 3100 Chipset is the concept of an error masking register. Each unit has a mask register, which blocks the recognition/logging/ reporting of each specific error type. Since the error will not be recognized when the corresponding mask bit is set, no error messages can be generated. This feature allows intelligent software to ignore specific error types during critical areas of code, where it does not want to be informed of errors that it will create, without ignoring other error types that it doesn't expect to happen. These mask bits will default to unmasked, and must be set by software or BIOS to take effect.

7.2.1.7.1 Locking DRAM Address and Syndrome on Errors

The first pair of error logging registers for CE (correctable errors) DRAM_SECF_ADD and DRAM_SECF_SYNDROME are locked when bit 0 of the DRAM_FERR is set. The second pair of error logging registers for CE (correctable errors) DRAM_SECN_ADD and DRAM_SECN_SYNDROME are locked when bit 0 of the DRAM_NERR is set. These pairs of two registers will retain their value even if new CE's are found. This allows the first (and possibly next) error to be captured and held instead of retaining the last. Corrected data errors as a result of either demand reads or scrubber-initiated traffic will be reflected in these error registers.

The logging register for UE (uncorrectable errors), DRAM_DED_ADD is locked when bit 1of the DRAM_FERR or DRAM_NERR is set. This register holds the address of uncorrectable errors on data reads not initiated by the scrubber for either periodic or demand scrubbing.

The logging register for Scrub detected errors, DRAM_SCRUB_ADD should be locked when bit 2 of the DRAM_FERR or DRAM_NERR is set. This register holds the address for scrubber-initiated transactions for either demand or periodic memory scrubbing.

The logging register for Retry detected errors, DRAM_RETRY_ADD is locked when bit 5 of the DRAM_FERR or DRAM_NERR is set. This register is locked when the determination is made that a retry to this address must be performed.

When the FERR/NERR registers are cleared the logging registers are free to update their contents until such time that either of these FERR/NERR registers again lock.

7.2.1.8 PCI Express* Errors and Errors on Behalf of PCI Express

IMCH-specific error detection, masking, and escalation mechanisms operate on a parallel path to their standardized counterparts included in the *PCI Express* Interface Specification, Rev 1.0a*. PCI Express errors are classified as either correctable or uncorrectable. Uncorrectable errors are further broken down as fatal or non-fatal.

PCI Express specified correctable errors are logged in the Correctable Error Status Register (Offset 110 - 113h: CORERRSTS – Correctable Error Status Register), unless they are masked by a corresponding bit in the Correctable Error Detect Mask Register (Offset 150 - 153h: COREDMASK – Correctable Error Detect Mask Register).

PCI Express specified uncorrectable errors are logged in the Uncorrectable Error Status Register (Offset 104 - 107h: UNCERRSTS – Uncorrectable Error Status Register), unless they are masked by a corresponding bit in the Uncorrectable Error Detect Mask Register (Offset 14C - 14Fh: UNCEDMASK – Uncorrectable Error Detect Mask Register). The Uncorrectable Error Severity Register (Offset 10C - 10Fh: UNCERRSEV – Uncorrectable Error Severity Register) determines if bits in the Uncorrectable Status register are treated as uncorrectable fatal or uncorrectable non-fatal errors. The Device Status register (6Eh) bits are set when the corresponding category of bit is set in the uncorrectable and correctable status registers.

Reporting of non-masked error bits to the root complex hierarchy of PCI Express error registers is controlled on three different levels. Individual errors are masked for reporting by the Uncorrectable Error Mask (Offset 108 - 10Bh: UNCERRMSK – Uncorrectable Error Mask Register) and the Correctable Error Mask (Offset 114 - 117h: CORERRMSK – Correctable Error Mask Register) registers. Individual error category (fatal, non-fatal, correctable, or unsupported) reporting is enabled in the Device Control Register (Offset 6C - 6Dh: PEADEVCTL – PCI Express Device Control Register) bits 3:0. Finally, uncorrectable error reporting (fatal or non-fatal) reporting may also be enabled by setting the SERR Enable bit in the PCI Command Register (Offset 04 - 05h: PCICMD – PCI Command Register).

There is an error pointer, in the Advanced Error Capability and Control Register (Offset 118 - 11Bh: AERCACR – Advanced Error Capabilities and Control Register) which will log the first uncorrectable error that is enabled for reporting. Also some uncorrectable errors, when they are the first uncorrectable error, will log their corresponding header log in the Header Log Registers (Offset 11C - 11Fh: HDRLOG0 – Header Log DW 0 (1st 32 bits) Register). An error pointer for unmasked correctable errors has been added in the Error Do Command Register (Offset 148 - 14Bh: PEAERRDOCMD – PCI Express Error Do Command Register).

These internally detected errors when they are reported are referred to as virtual error messages. These are different from errors which are detected by the downstream device which then sends an error message to the root complex, which are referred to as externally detected or “received’ error messages. The received system error bit in the Secondary Status Register (Offset 1E - 1Fh: SECSTS – Secondary Status Register) is set when either fatal or non-fatal messages are received at the root complex.

At this point in the PCI Express error hierarchy, these virtual error messages are logically ORed with the received error messages, and will just be referred to as fatal, non-fatal, or correctable error messages, no reference to either virtual or received.

When enabled by the enable system error bit in the PCI Command Register, any fatal or non-fatal messages will set the signaled system error bit in the PCI Status register (Offset 06 - 07h: PCISTS – PCI Status Register). The Root Port Error Message Status Register (Offset 130 - 133h: RPERRMSTS – Root (Port) Error Message Status Register) will indicate first and multiple errors of each error message category, and the corresponding error source IDs of the first correctable and uncorrectable error messages will be the logged in the Error Source ID register (134h).

These errors that have been reported to the root complex can now be reported to the system, via the category enables in the Root Port Error Command Register (Device 2-3, Function 0, Offset 12C-12Fh) for interrupts. These interrupts can be in the form of legacy type interrupts if enabled in the PCI command register and MSI is not enabled, or message signaled interrupts if enabled in the MSI Capabilities register (Offset 5A - 5Bh: MSICAPA – MSI Capabilities Register).

The Root Port Control register (Offset 80 - 83h: PEARPCTL – PCI Express Root Port Control Register) enables errors to be reported to the system via other IMCH specific methods, again on a category basis. The Error Do Command register, selects between the four methods of system signaling, SERR, SCI, SMI, and MCERR.

The error model outside of PCI Express includes a local FERR/NERR pair of registers in each unit and a global FERR/NERR pair of registers that indicates which unit had problems. The Local FERR/NERR register pair (Offset 160 - 163h: PEAFERR – PCI Express First Error Register and Offset 80 - 83h: PEARPCTL – PCI Express Root Port Control Register) includes PCI Express defined errors and additional detected errors within the PCI Express unit. This register pair has three sets of error bits for the three categories of errors: the first set for received messages, the second set for internally detected errors (virtual messages need not have been generated), and unit specific errors outside of the PCI Express spec, and the third set for device errors. This error scheme sets FERR/NERR error bits regardless whether or not they were reported via interrupt or other signaling method.

The signaling due to unit specific errors has its logic dependent on the PCI Express Unit Error Register (Offset 140 - 143h: PEAUNITERR – PCI Express Unit Error Register). The errors flagged in this register must be cleared before exiting the error service routine.

The signaling due to received messages has its logic dependent on the Root Error Message Status register (Offset 130 - 133h: RPERRMSTS – Root (Port) Error Message Status Register). The Root Error Status register must be cleared before exiting the error service routine.

The signaling due to internally detected PCI Express errors has its logic dependent on the Device Status register (Offset 6E - 6Fh: PEADEVSTS – PCI Express Device Status Register). The Device Status register must be cleared before exiting the error service routine.

Software must clear the global FERR first, and then the global NERR. Software then clears the local FERR register and the local NERR register of each unit in that order. After clearing FERR and then clearing NERR, the local FERR must be read to make sure that it remains ‘0’ indicating no more errors have occurred during the clearing of these registers. After all units’ FERR & NERR registers have been cleared, the global FERR is again read to ensure that no additional errors occurred during the clearing sequence.

Since the PCI Express units have more hierarchy than other units, more registers must be cleared other than just the local FERR and NERR registers. After clearing the local FERR & NERR, one must also clear the Root Error Status, Unit Error Status, Device Status, Uncorrectable Error Status, and Correctable Error Status registers. One only needs to clear the PCI Status and Secondary Status registers if these are being utilized in a given particular error model. No logic depends on the state of any of these status bits. If not utilized, they can be ignored.

PCI Express Root Port Device, Uncorrectable Error Status, the PCI Express Root Port Device, Correctable Error Status, the PCI Express Root Port Device, Error Status, the

7.2.1.9 Configurable Error Containment at the Legacy Interface

Depending on the I/O devices in use, data errors could have catastrophic effects when allowed to propagate. The Legacy interface has the configurability of allowing the poisoning and propagation of data errors or to stop the data from transferring at all and escalate the data errors to the system. This is extreme behavior, which can be enabled or disabled, in order to prevent data corruption on a critical device, and is referred to as “stop and scream”.

7.3 Error Conditions Signaled

Intel® 3100 Chipset notification action taken upon detection of an error is controlled through three registers. The SERRCMD register enables the generation of the SERR message, the SCICMD register enables the generation of the SCI message, and the SMICMD register enables the generation of SMI messages. Special cycle types of DO_SERR, DO_SCI, or DO_SMI may be transmitted to notify the processor of the condition.

Once the processor has been interrupted, it polls the system to determine the cause of the exception. If the IMCH initiated the exception condition by sending a message over NSI, then the processor is so informed by the IICH. At this point, the processor may read the IMCH's error status registers to determine the exact cause of the condition. The processor explicitly clears the status bit that points to the exception condition.

The IMCH in addition to signaling errors to the IICH for further handling, has added the capability of signaling the processor directly by use of the front side bus error signal MCERR#. The processor, upon observing this signal active, enters into special error handling code known as machine check code.

For each type of error detected in a given unit, there is a bit that corresponds to that error in the unit_FERR, unit_NERR, SERRCMD_unit, SMICMD_unit, SCICMD_unit, and MCERRCMD_unit registers. (Note that one and only one xCMD bit can be enabled per error type.) The first occurrence of an error type will be indicated by the bit assertion in the unit_FERR. If that error occurs again than the corresponding bit will be set in the unit_NERR register. When a bit is asserted in either the unit_FERR or unit_NERR, and if the corresponding enable bit is set in one of the named CMD registers, then an error signal will be asserted, corresponding to the name of the CMD register: DO_SERR, DO_SCI, DO_SMI, or DO_MCERR. The assertion of the DO_SERR signal also requires that the SERR enable in the PCICMD register is set. The assertion of the DO_SERR signal also causes the appropriate SERR signaled status bit to be set in the PCISTS register.

Table 106. Pseudocode for EDMA Errors (Sheet 1 of 2)

Condition	Source	Action	Status
The descriptor pointer in next descriptor address register is of incorrect type or range for EDMA channel 3.	Internal	**DO_SERR** and set PCISTS10[SSE] if (PCICMD10[SERRE]=1 AND EDMA_EMASK[7]=0 AND SERRCMD_EDMA[7]=1 AND R_EDGE{EDMA_FERR[31] OR EDMA_NERR[31]}); **DO_SMI** if EDMA_EMASK[7]=0 AND SMICMD_EDMA[7]=1 AND R_EDGE{EDMA_FERR[31] OR EDMA_NERR[31]}; **DO_SCI** if EDMA_EMASK[7]=0 AND SCICMD_EDMA[7]=1 AND R_EDGE{EDMA_FERR[31] OR EDMA_NERR[31]}; **DO_MCERR** if EDMA_EMASK[7]=0 AND **MC**ERRCMD_EDMA[7]=1 AND R_EDGE{EDMA_FERR[31] OR EDMA_NERR[31]};	EDMA_FERR[31]
The descriptor pointer in next descriptor address register is not aligned to eight double-word boundary for EDMA channel 3.	Internal	**DO_SERR** and set PCISTS10[SSE] if (PCICMD10[SERRE]=1 AND EDMA_EMASK[6]=0 AND SERRCMD_EDMA[6]=1 AND R_EDGE{EDMA_FERR[30] OR EDMA_NERR[30]}); **DO_SMI** if EDMA_EMASK[6]=0 AND SMICMD_EDMA[6]=1 AND R_EDGE{EDMA_FERR[30] OR EDMA_NERR[30]}; **DO_SCI** if EDMA_EMASK[6]=0 AND SCICMD_EDMA[6]=1 AND R_EDGE{EDMA_FERR[30] OR EDMA_NERR[30]}; **DO_MCERR** if EDMA_EMASK[6]=0 AND **MC**ERRCMD_EDMA[6]=1 AND R_EDGE{EDMA_FERR[30] OR EDMA_NERR[30]};	EDMA_FERR[30]
The source address does not comply with the source type or range for EDMA channel 3.	Internal	**DO_SERR** and set PCISTS10[SSE] if (PCICMD10[SERRE]=1 AND EDMA_EMASK[5]=0 AND SERRCMD_EDMA[5]=1 AND R_EDGE{EDMA_FERR[29] OR EDMA_NERR[29]}); **DO_SMI** if EDMA_EMASK[5]=0 AND SMICMD_EDMA[5]=1 AND R_EDGE{EDMA_FERR[29] OR EDMA_NERR[29]}; **DO_SCI** if EDMA_EMASK[5]=0 AND SCICMD_EDMA[5]=1 AND R_EDGE{EDMA_FERR[29] OR EDMA_NERR[29]}; **DO_MCERR** if EDMA_EMASK[5]=0 AND **MC**ERRCMD_EDMA[5]=1 AND R_EDGE{EDMA_FERR[29] OR EDMA_NERR[29]};	EDMA_NERR[29]
The source address is not aligned as specified by the source address bit for EDMA channel 3.	Internal	**DO_SERR** and set PCISTS10[SSE] if (PCICMD10[SERRE]=1 AND EDMA_EMASK[4]=0 AND SERRCMD_EDMA[4]=1 AND R_EDGE{EDMA_FERR[28] OR EDMA_NERR[28]}); **DO_SMI** if EDMA_EMASK[4]=0 AND SMICMD_EDMA[4]=1 AND R_EDGE{EDMA_FERR[28] OR EDMA_NERR[28]}; **DO_SCI** if EDMA_EMASK[4]=0 AND SCICMD_EDMA[4]=1 AND R_EDGE{EDMA_FERR[28] OR EDMA_NERR[28]}; **DO_MCERR** if EDMA_EMASK[4]=0 AND **MC**ERRCMD_EDMA[4]=1 AND R_EDGE{EDMA_FERR[28] OR EDMA_NERR[28]};	EDMA_FERR[28]
The destination address does not comply with the destination type or range for EDMA channel 3.	Internal	**DO_SERR** and set PCISTS10[SSE] if (PCICMD10[SERRE]=1 AND EDMA_EMASK[3]=0 AND SERRCMD_EDMA[3]=1 AND R_EDGE{EDMA_FERR[27] OR EDMA_NERR[27]}); **DO_SMI** if EDMA_EMASK[3]=0 AND SMICMD_EDMA[3]=1 AND R_EDGE{EDMA_FERR[27] OR EDMA_NERR[27]}; **DO_SCI** if EDMA_EMASK[3]=0 AND SCICMD_EDMA[3]=1 AND R_EDGE{EDMA_FERR[27] OR EDMA_NERR[27]}; **DO_MCERR** if EDMA_EMASK[3]=0 AND **MC**ERRCMD_EDMA[3]=1 AND R_EDGE{EDMA_FERR[27] OR EDMA_NERR[27]};	EDMA_FERR[27]
The destination address is not aligned as specified by the destination address bit for EDMA channel 3.	Internal	**DO_SERR** and set PCISTS10[SSE] if (PCICMD10[SERRE]=1 AND EDMA_EMASK[2]=0 AND SERRCMD_EDMA[2]=1 AND R_EDGE{EDMA_FERR[26] OR EDMA_NERR[26]}); **DO_SMI** if EDMA_EMASK[2]=0 AND SMICMD_EDMA[2]=1 AND R_EDGE{EDMA_FERR[26] OR EDMA_NERR[26]}; **DO_SCI** if EDMA_EMASK[2]=0 AND SCICMD_EDMA[2]=1 AND R_EDGE{EDMA_FERR[26] OR EDMA_NERR[26]}; **DO_MCERR** if EDMA_EMASK[2]=0 AND **MC**ERRCMD_EDMA[2]=1 AND R_EDGE{EDMA_FERR[26] OR EDMA_NERR[26]};	EDMA_FERR[26]

Table 106. Pseudocode for EDMA Errors (Sheet 2 of 2)

Condition	Source	Action	Status
Data parity Error in reading source data from system memory for EDMA channel 3.	Internal	**DO_SERR** and set PCISTS10[SSE] if (PCICMD10[SERRE]=1 AND EDMA_EMASK[1]=0 AND SERRCMD_EDMA[1]=1 AND R_EDGE{EDMA_FERR[25] OR EDMA_NERR[25]}); **DO_SMI** if EDMA_EMASK[1]=0 AND SMICMD_EDMA[1]=1 AND R_EDGE{EDMA_FERR[25] OR EDMA_NERR[25]}; **DO_SCI** if EDMA_EMASK[1]=0 AND SCICMD_EDMA[1]=1 AND R_EDGE{EDMA_FERR[25] OR EDMA_NERR[25]}; **DO_MCERR** if EDMA_EMASK1]=0 AND **MC**ERRCMD_EDMA1]=1 AND R_EDGE{EDMA_FERR[25] OR EDMA_NERR[25]};	EDMA_FERR[25]
Received configuration /TAP write command when EDMA is in Normal Mode for EDMA channel 3.	Internal	**DO_SERR** and set PCISTS10[SSE] if (PCICMD10[SERRE]=1 AND EDMA_EMASK[0]=0 AND SERRCMD_EDMA[0]=1 AND R_EDGE{EDMA_FERR[24] OR EDMA_NERR[24]}); **DO_SMI** if EDMA_EMASK[0]=0 AND SMICMD_EDMA[0]=1 AND R_EDGE{EDMA_FERR[24] OR EDMA_NERR[24]}; **DO_SCI** if EDMA_EMASK[0]=0 AND SCICMD_EDMA[0]=1 AND R_EDGE{EDMA_FERR[24] OR EDMA_NERR[24]}; **DO_MCERR** if EDMA_EMASK[0]=0 AND **MC**ERRCMD_EDMA[0]=1 AND R_EDGE{EDMA_FERR[24] OR EDMA_NERR[24]};	EDMA_FERR[24]
EDMA channel 2 errors	Internal	Same bit functionality as bits 31:24 except these are for EDMA channel 2. (Use bits 23:16)	
EDMA channel 1 errors	Internal	Same bit functionality as bits 31:24 except these are for EDMA channel 1. (Use bits 15:8)	
EDMA channel 0 errors	Internal	Same bit functionality as bits 31:24 except these are for EDMA channel 0. (Use bits 7:0)	

8.0 Platform Management (IMCH)

This chapter provides an overview of the system management support provided by the IMCH. There are three primary management support features in the IMCH.

1. Integrated system management bus (SMBus) interface
2. Hot-plug capability on the IMCH PCI Express* interface
3. Architectural support for platform power management

Note: Material in this chapter is specific to the IMCH and does not apply to the IICH.

8.1 Integrated SMBus Interface

The IMCH provides a fully functional System Management Bus (SMBus) target interface, which provides direct access to all internal IMCH configuration register space through SMBSDA and SMBSCL. SMBus access is available to all internal configuration registers, regardless of whether the register in question is normally accessed via the memory-mapped mechanism or the standard configuration mechanism. This provides for highly flexible platform management architectures, particularly given a baseboard management controller (BMC) with an integrated network interface controller (NIC) function.

For a full behavioral description of the IICH SMBus (SMBDATA and SMBCLK), see Chapter 26.0, “Device 31, Function 3: SMBus Controller Functional Description.”

8.2 SMBus Target Architecture

The SMBus target integrated into the IMCH is compatible with revision 2.0 of the *SMBus Specification*. A brief overview of the SMBus architecture is provided here for reference.

8.2.1 High Level Operation

The SMBus interface consists of two interface pins: clock and serial data. Multiple initiator and target devices may be electrically present on the same pair of signals. Each target recognizes a start signaling semantic, and recognizes its own seven-bit address to identify pertinent bus traffic. The IMCH address is hard-coded to 0110_000. The least significant bit, read/write bit, will be appended. The definition of least significant bit is in the *SMBus Specification*.

The protocol also allows for traffic to stop in “midsentence,” requiring all targets to tolerate and properly “clean up” in the event of an access sequence that is abandoned by the initiator prior to normal completion. The IMCH is compliant with this requirement.

Additionally, the protocol comprehends “wait states” on read and write operations, which the IMCH takes advantage of to keep the bus busy during internal configuration space accesses.

8.2.1.1 SMBus Register Summary

Table 107 provides a quick-reference summary of the SMBus target register space. These registers are part of the target itself and therefore not accessible by any other means other than the direct SMBus connection.

Table 107. SMBus Register Summary

Symbol	Full Name/Function
CMD	Command
BYTCNT	Byte Count
ADDR3	Bus Number (Only lower five bits are utilized)
ADDR2	Device/Function Number
ADDR1	Extended Reg Number (Bits 03:00 - 4 k page extension)
ADDR0	Register Number (offset into function space)
DATA3	Data, fourth byte (31:24)
DATA2	Data, third byte (23:16)
DATA1	Data, second byte (15:08)
DATA0	Data, first byte (07:00)
STS	Status, only for reads

Table 108. SMBUS Memory-Mapped Register Summary

Symbol	Full Name/Function
CMD	Command
BYTCNT	Byte Count
ADDR3	Destination Memory (BAR Selection)
ADDR2	Address Offset 23:16 (Filler-used to zero out register)
ADDR1	Address Offset 15:08 (15:12 not used)
ADDR0	Address Offset 07:00 (11:00 used for 4 K page)
DATA3	Data, fourth byte (31:24)
DATA2	Data, third byte (23:16)
DATA1	Data, second byte (15:08)
DATA0	Data, first byte (07:00)
STS	Status, only for reads

Table 109. ADDR3 Memory Assignments

ADDR3	Destination Memory Assignments
00_000001	EDMA
00_001000	DDR2
All others	Reserved

Table 107 and Table 108 indicate the sequence of data as it is presented on the SMBUS following the byte address of the IMCH itself. This is not to indicate any specific register stack or array implemented in the IMCH. The registers can take on different meanings

depending on whether it is a configuration or memory-mapped access type. The command indicates how to interpret the registers. Refer to the *SMBus Interface Specification Rev. 2.0* for interface protocol details.

8.2.1.2 Internal Register Access Mechanism

All SMBus accesses to internal register space are initiated via a write to the CMD register. Any register writes received by the IMCH while a command is already in progress receive a NAK to prevent spurious operation. The master is no longer expected to poll the CMD register to prevent clobbering a command in progress prior to issuing further writes. The SMBus access is delayed by stretching the clock until such time that the data is delivered. Note that per the *SMBus Interface Specification Rev. 2.0*, this can not be longer than 25 ms. To set up an internal access, the four ADDR bytes are programmed followed by a command indicator to execute a read or write. Depending on the type of access, these four bytes indicate either the Bus number, Device, Function, Extended Register Offset, and Register Offset, or the Memory-mapped region selected and the address within the region. The configuration type access utilizes the traditional bus number, device, function, and register offset; but in addition, also uses an extended register offset which expands the addressable register space from 256 bytes to 4 Kbyte. The memory-mapped type access redefines these bytes to be a memory-mapped region selection byte and the memory address within the region. Refer to Table 108 and Table 109 which display this information.

FSB-initiated accesses to registers are serviced through the configuration ring. For these registers, it is perfectly legal for an SMBus access to be requested while an FSB-initiated access is already in progress. The IMCH supports “wait your turn” arbitration to resolve all collisions and overlaps, such that the access that reaches the configuration ring arbiter first is serviced first while the conflicting access is held off. An absolute tie at the arbiter is resolved in favor of the FSB.

8.2.1.3 SMBus Register Definitions

8.2.1.3.1 CMD - Command Register

When written, this Command Register indicates the type and size of transfer. All configuration accesses from the SMBus port are initiated by writing to this register. While a command is in progress, all future writes or reads are NACK’d by the IMCH to avoid having registers overwritten while in use. The two command size fields allows for more flexibility on how the data payload is transferred, both internally and externally. The begin and end bits support the breaking of the transaction up into smaller transfers, by defining the start and finish of an overall transfer.

Table 110. Command (CMD) Register (Sheet 1 of 2)

Bit	Description
07	Begin Transaction Indicator 0 = Current transaction is NOT the first of a read or write sequence. 1 = Current transaction is the first of a read or write sequence. On a single transaction sequence this bit is set along with the End Transaction Indicator.
06	End Transition Indicator 0 = Current transaction is NOT the last of a read or write sequence. 1 = Current transaction is the last of a read or write sequence. On a single transaction sequence this bit is set along with the Begin Transaction Indicator.
05	Address Mode: Indicates whether memory or configuration space is being accessed in this SMBus sequence. 0 = Memory Mapped Mode 1 = Configuration Register Mode

Table 110. Command (CMD) Register (Sheet 2 of 2)

Bit	Description
04	Reserved - Set to 0.
03:02	Internal Command Size: All accesses are naturally aligned to the access width. This field specifies the internal command to be issued by the SMBus slave logic to the IMCH core. 00 = Read Dword 01 = Write Byte 10 = Write Word 11 = Write Dword
01:00	SMBus Command Size: This field specifies the SMBus command to be issued on the SMBus. This field is used as an indication of the length of the transfer so that the slave knows when to expect the PEC packet (if enabled). 00 = Byte 01 = Word 10 = Dword 11 = Reserved

8.2.1.3.2 BYTCNT - Byte Count Register

The byte count register indicates the number of bytes following the byte count register when performing a write or when setting up for a read. The byte count is also used when returning data to indicate the following number of bytes (including the status byte) which are returned prior to the data. Note that the byte count is only transmitted for block type accesses on SMBus. SMBus word or byte accesses do not use the byte count.

Table 111. Byte Count Register

Position	Description
07:00	Byte Count: Number of bytes following the byte count for a transaction.

8.2.1.3.3 ADDR3 – Address Byte 3 Register

This register must be programmed with the Bus Number of the desired configuration register in the lower five bits for a configuration access. For a memory-mapped access this field selects which memory-map region is being accessed. There is no status bit to poll to see if a transfer is currently in progress, because by definition, if the transfer completed, the task is done. The clock stretch is used to guarantee the transfer is truly complete.

The Intel® 3100 Chipset does not support access to other logical bus numbers via the SMBus port. All registers “attached” to the configuration mechanism that the SMBus has access to, reside on logical bus#0.

Table 112. Address Byte 3 Register

Position	Configuration Register Mode Description	Memory Mapped Mode Description
07:05	Ignored.	Memory map region to access. 01h = EDMA 08h = DDR2 Others = Reserved
04:00	Bus Number: Must be zero: the SMBus port can only access devices on the IMCH and all devices are bus zero.	

8.2.1.3.4 ADDR2 – Address Byte 2 Register

This register must be programmed with the Device Number and Function Number of the desired configuration register for a configuration type access, otherwise it must be set to zero.

Table 113. ADDR2 – Address Byte 2 Register

Position	Configuration Register Mode Description	Memory Mapped Mode Description
07:03	Device Number. Can only be devices on the IMCH.	Zeros used for padding.
02:00	Function Number.	

8.2.1.3.5 ADDR1 – Address Byte 1 Register

This register must be programmed with the upper address bits for the register with the 4K region. Whether it is a configuration or memory-map type of access, only the lower bits are utilized, the upper four bits are ignored.

Table 114. ADDR1 – Address Byte 1 Register

Position	Description
07:04	Ignored.
03:00	Extended Register Number. Upper address bits for the 4 K region of register offset.

8.2.1.3.6 ADDR0 – Address Byte 0 Register

This register indicates the lower eight address bits for the register within the 4 K region, regardless whether it is a configuration or memory-map type of access.

Table 115. ADDR0 – Address Byte 0 Register

Position	Description
07:00	Register Offset.

8.2.1.3.7 Offset 04 - 07h: DATA – Data Register

This field is used to receive read data or to provide write data associated with the desired register.

At the completion of a read command, this field contains the data retrieved from the selected register. All reads return an entire aligned Dword (32 bits) of data.

The appropriate number of byte(s) of this 32 bit logical register must be written with the desired write data prior to issuing a write command. For a byte write only bits 07:00 are used, for a Word write only bits 15:0 are used, and for a Dword write all 32 bits are used.

Table 116. Offset 04-07: DATA - Data Register

Bits	Type	Reset	Description
31:24	RW	00h	**Byte 3 (DATA3):** Data bits [31:24]
23:16	RW	00h	**Byte 2 (DATA2):** Data bits [23:16]
15:08	RW	00h	**Byte 1 (DATA1):** Data bits [15:8]
07:00	RW	00h	**Byte 0 (DATA0):** Data bits [7:0]

8.2.1.3.8 STS – Status Register

For a read cycle, the data is preceded by a byte of status. Table 117 shows how these bits are defined.

Table 117. Status Register

Position	Description
07	Internal Timeout. 0 = SMBus request is completed within 2 ms internally 1 = SMBus request is not completed in 2 ms internally
06	Ignored.
05	Internal Master Abort. 0 = No Internal Master Abort Detected 1 = Detected an Internal Master Abort
04	Internal Target Abort. 0 = No Internal Target Abort Detected 1 = Detected an Internal Target Abort
03:01	Ignored.
00	Successful. 0 = The last SMBus transaction was not completed successfully 1 = The last SMBus transaction was completed successfully

8.2.1.4 Unsupported Access Addresses

It is possible for an SMBus master to program an unsupported bit combination into the ADDR registers. The IMCH does not support such usage, and may not gracefully terminate such accesses.

8.2.1.5 SMB Transaction Pictograms

Since the new SMB target interface is of enterprise origin, it is more complex than the original SMB target interface of desktop origin. The following drawings are included to demonstrate the different types of transactions, especially how they can be broken up into multiple smaller transfers.

Figure 54. Dword Configuration Read Protocol

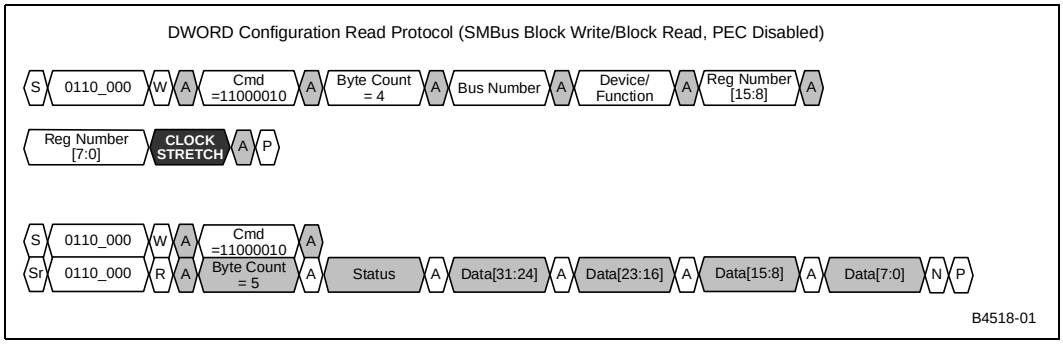


Figure 55. Dword Configuration Write Protocol

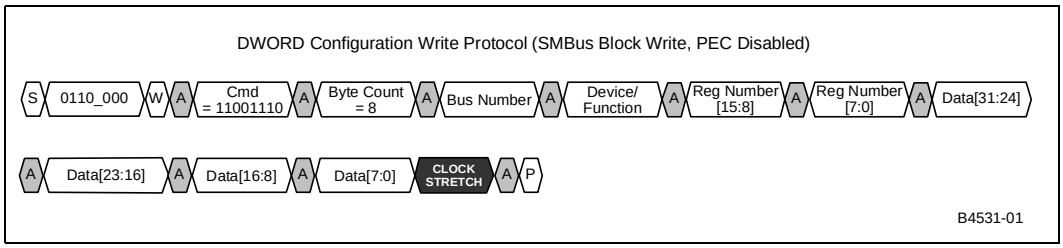


Figure 56. Dword Memory Read Protocol

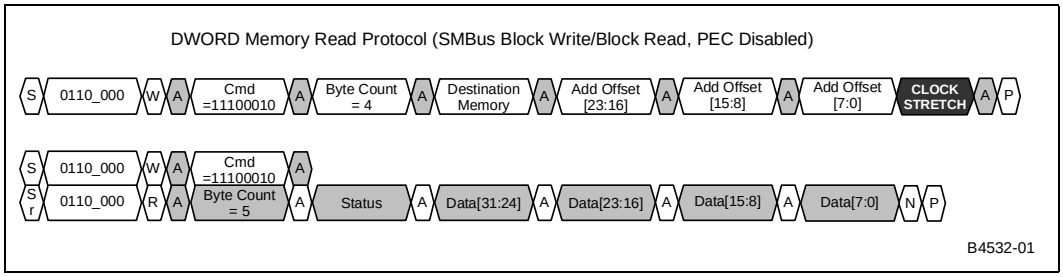


Figure 57. Dword Memory Write Protocol

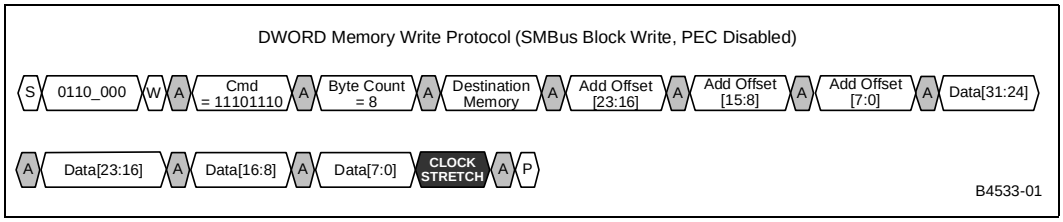


Figure 58. Dword Configuration Read Protocol

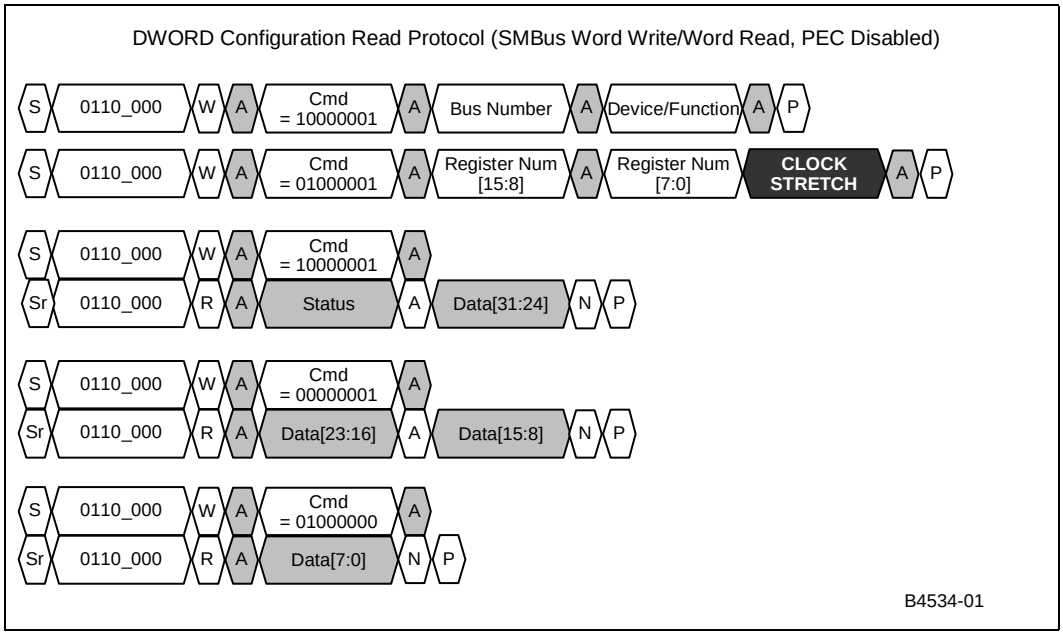


Figure 59. Dword Configuration Write Protocol

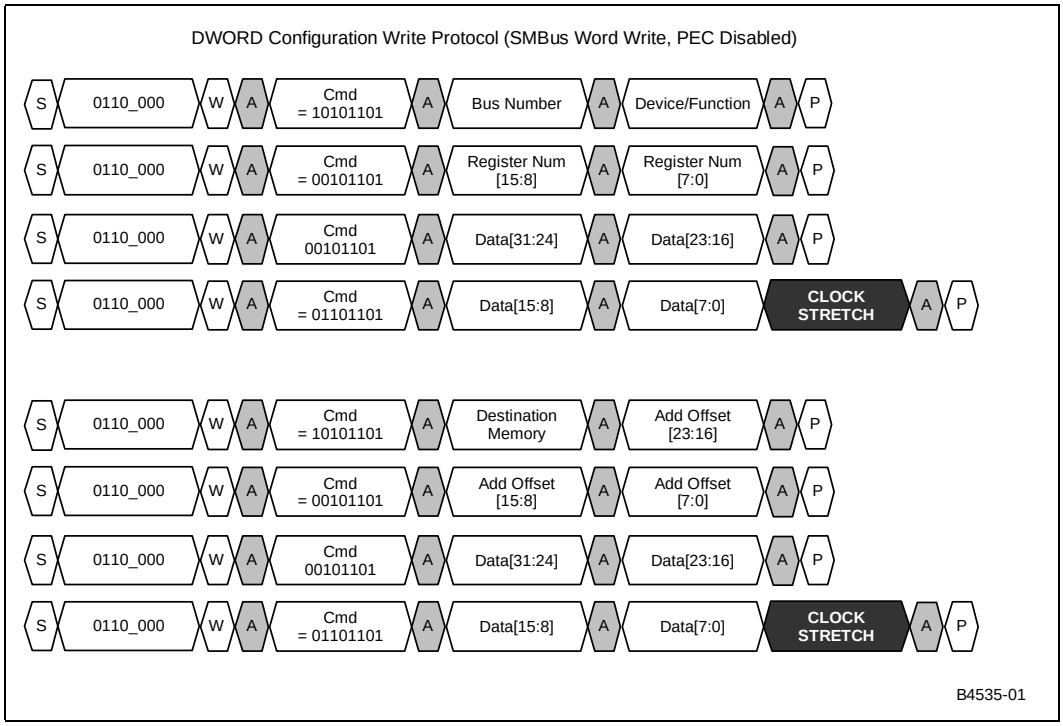


Figure 60. Dword Memory Read Protocol

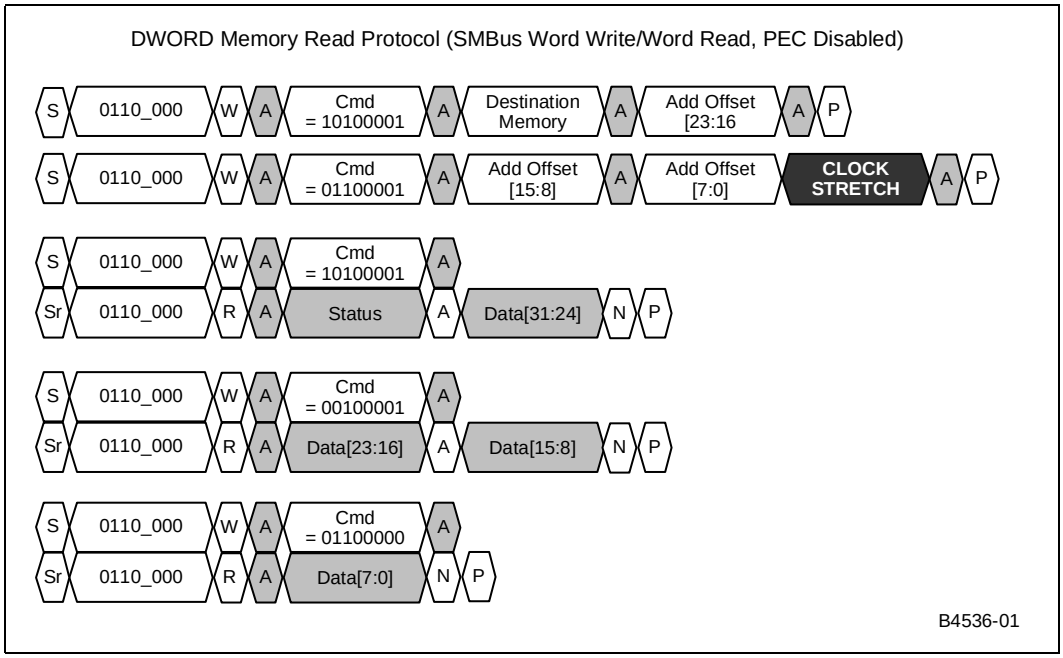
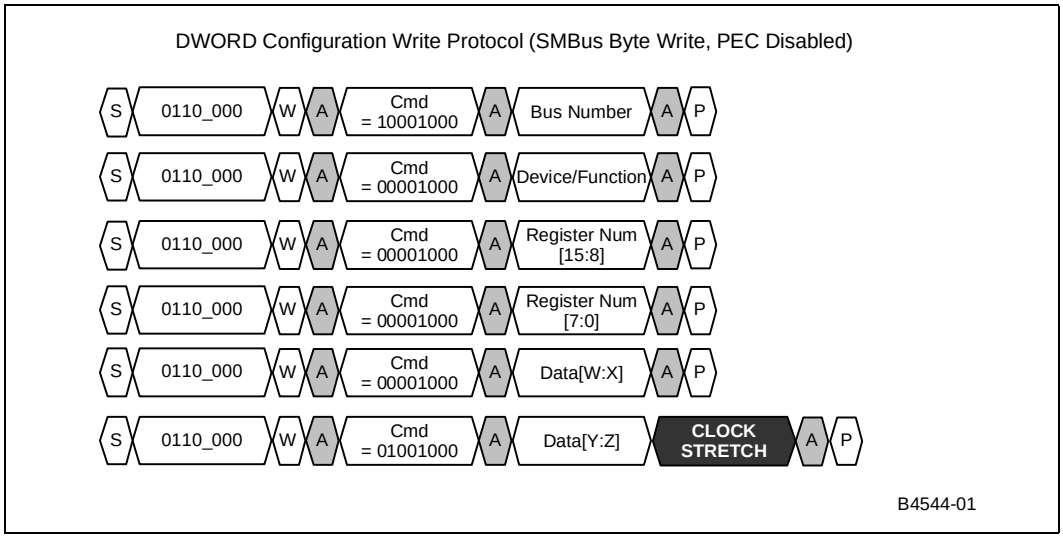


Figure 61. WORD Configuration Write Protocol



8.2.2 Suggested SMBus Usage Models

8.2.2.1 Remote Error Handling

The Intel® 3100 Chipset supports error escalation via both SMI and MCERR FSB signaling, thus error handling may be implemented in system management mode (SMM) software, machine check architecture (MCA) software, or a combination of the two. Such software could direct a BMC with an integrated NIC to “call home” when errors are reported by the IMCH. The BMC could then interrogate internal IMCH error logging registers under remote control across the network interface, providing full identification and isolation of reported errors as described elsewhere in this document. The further possibility exists for remotely managed reconfiguration via the SMBus target port, as well as remotely managed system reboot via the BMC (if necessary).

8.2.2.2 Remote Platform Monitoring

The SMBus target also provides a sophisticated BMC the capability to monitor the health of a Intel® 3100 Chipset-based platform, such that statistics on correctable error location and frequency may be tracked remotely in an effort to anticipate and prevent more serious failures.

The IMCH includes significant RASUM functionality on both its memory subsystem and its PCI Express interfaces. Some types of errors are expected at a modest frequency within a platform of this complexity, and the IMCH provides internal hardware to track the frequency of such errors. These include correctable ECC errors on the memory interface, as well as transient communication errors on the high-speed serial PCI Express interfaces – refer to Chapter 7.0, “RAS Features and Exception Handling” for further details.

The BMC could be directed remotely to periodically poll the internal error logging registers of the IMCH, permitting a remote management software package to maintain a running profile of error types and frequencies experienced by a Intel® 3100 Chipset-based platform. Changes in error frequency or type could be flagged by the remote monitoring software to prompt follow-up preventative maintenance on the platform.

8.3 PCI Express* Hot-Plug Capability

The IMCH implements the optional hot-plug capability on its PCI Express interfaces. PCI Express native Hot-Plug allows for higher availability and serviceability. It gives the user the capability of adding, removing, or swapping out a PCI Express slot device without powering down the system. The user and system communicate through a combination of software and hardware utilizing notification through mechanical means and indicator lights. The signals involved are briefly discussed here. Refer to the *PCI Express* Interface Specification, Rev. 1.0a* for further details.

8.3.1 Architectural Support for Hot-Plug

Support for PCI Express level hot-plug is basically divided into two parts. First, the IMCH supports a standard software interface for hot-plug capability management; defined for PCI and propagated to PCI Express. Second, the IMCH supports a mechanism to provide the external hardware connectivity required to implement the standard register interface.

8.3.1.1 Hot-Plug Software Interface

Not all concepts from the Standard PCI Hot Plug definition apply directly to PCI Express interfaces. The specification still calls for an identical software interface in order to facilitate adoption with minimal development overhead on this aspect of the implementation.

The largest variance from the legacy PCI hot-plug model is in control of the interface itself. PCI required arbitration support for idling already connected components, and “quick switches” to isolate the bus interface pins of a hot-plug slot. PCI Express is a point-to-point interface, making hot-plug a degenerate case of the old model that doesn’t require such arbiter support. Furthermore, the PCI Express interface is inherently tolerant of hot connect or disconnect, and does not have explicit clock or reset pins defined as a part of the bus, although they are standard pieces of some defined PCI Express connector form factors. As a result of these differences, some of the inherited hot plug command and status codes are misleading when applied to PCI Express.

The Hot-Plug function (when available) is advertised in the capabilities list for its associated virtual P2P bridge device within the IMCH. When hot-plug is unavailable, the capabilities list does not appear, preventing spurious detection by software.

The compatible set of hot-plug registers may be accessed via the IMCH configuration mechanism as defined in the configuration chapter of this document.

For specific information on the hot-plug register set, refer to the Chapter 2.0, “Configuration Register Descriptions.”

8.3.1.2 PCI Express* Hot-Plug Overview

PCI Express native Hot-Plug allows for higher availability and serviceability. It gives the user the capability of adding, removing, or swapping out a PCI Express slot device without taking down the system. The user and system communicate through a combination of software and hardware utilizing notification through mechanical means and indicator lights. The signals involved are briefly discussed here. Refer to the *PCI Express Specification* for complete details.

The *PCI Express Specification* refers to two major form factors. PCI Express-C is a traditional card-edge connector slot, which is supported by the Intel® 3100 Chipset. PCI Express-M utilizes a modular form factor connector that is not supported by the Intel® 3100 Chipset.

The PCI Express hot-plug register group includes attention button, attention indicator, power controller, power indicator, presence detect, and mechanical retention latch. Of particular interest to this discussion is what signals are necessary to support PCI Express hot-plug. The Attention Button Registers require an input from an external attention button. The Attention Indicator Registers utilize an output to drive an external indicator light. The Power Indicator Registers utilize an output to drive an external power indicator light. The Presence Detect Registers require an input pin to detect the presence of an add-in card. The MRL or mechanical retention latch requires an additional input from some sort of sensor on the retention mechanism. The Power Controller Register requires an output to control the power supplied to the add-in card. It also requires an input to indicate a power fault when it occurs. The total of all these signals comes to four inputs (attention pushbutton, presence detect, power fault, MRL sensor) and three outputs (attention indicator, power indicator, and power enable/control) for a total of seven pins. Normally these seven pins would come directly off of the integrated hot-plug controller, but a different approach has been taken to support PCI Express hot-plug.

The Intel® 3100 Chipset utilizes an external I/O expander device, which interfaces to the SMBus. As a result, the Intel® 3100 Chipset provides both a master and a target on its SMBus interface. This allows the IMCH to control an external device that in turn provides the interface pins required to support PCI Express hot plug. The IMCH's PE_HPINTR# pin support a hot-plug external devices EXTINTR#, which indicates that one of its input pins have changed.

Figure 62 is a generic diagram of a PCI Express hot-plug implementation. The Intel® 3100 Chipset only supports hot-plug on the 0 side of the PCI Express A port (PEA).The non-hot-plug or traditional interface signals have a direct path to the slot. The hot-plug signals communicate with a common hot-plug controller. This controller then interfaces to the SMBus, which connects to the external I/O expander. It is the external expander that interfaces directly with the hot-plug signals of the slot. The "slot" reference is a generic one, since not all hot-plug signals actually go to the card, but to logic that supports the slot. In fact, only the presence detect signal come directly from the actual card slot. The power controller, pushbutton, and indicator lights associated with each slot are not on the actual add-in card.

The diagram in Figure 62 separates the IMCH from external components. The external features include the slots themselves, the I/O expander device, a pull-up resistor on the external interrupt, debounce-circuitry for the attention buttons (designated as DB), and board loop-back for the spare output pin for validation purposes. The BMC and PXH could be other components resident on the SMBus as example.

Figure 62. PCI Express* Hot-Plug Overview

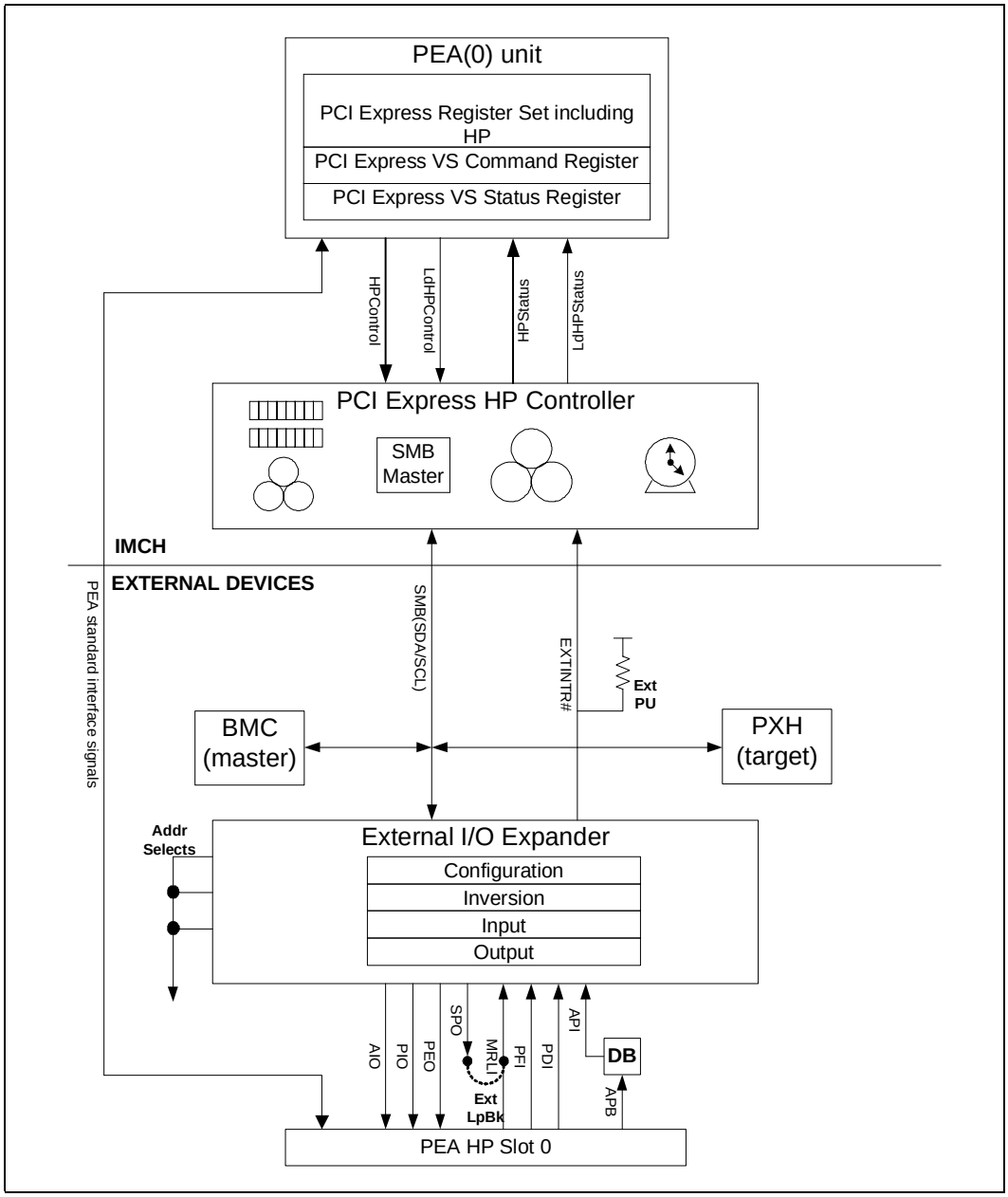


Table 118 defines the IMCH PCI Express Hot Plug slot signals shown in Figure 62.

Table 118. Slot Signal Legend

API – Attention Pushbutton Input	AIO – Attention Indicator Output
PDI – Presence Detect Input	PIO – Power Indicator Output
PFI – Power Fault Input	PEO – Power Enable Output
MRLI – MRL Input	SPO – Spare Output

8.3.1.3 Hot-Plug Controller Flow Diagrams

The hot-plug controller tasks include:

- Configuring the external I/O expander device at boot time under the control of BIOS.
- Monitoring the expander inputs via its interrupt pin and presenting changes to the PCI Express unit.
- Updating the expander registers based on writes to the PCI Express unit configuration registers.
- Blinking the attention and power indicators based on the PCI Express unit register bits.
- Interfacing to the SMB via the SMB master.

The following three flowcharts have been created to provide a more compete understanding of the Hot-Plug sequence of events. Figure 63 addresses hot-plug initialization at boot time. Figure 64 addresses the insertion of a card after initialization. Figure 65 addresses the removal of a card.

Figure 63. Hot-Plug Initialization Flow

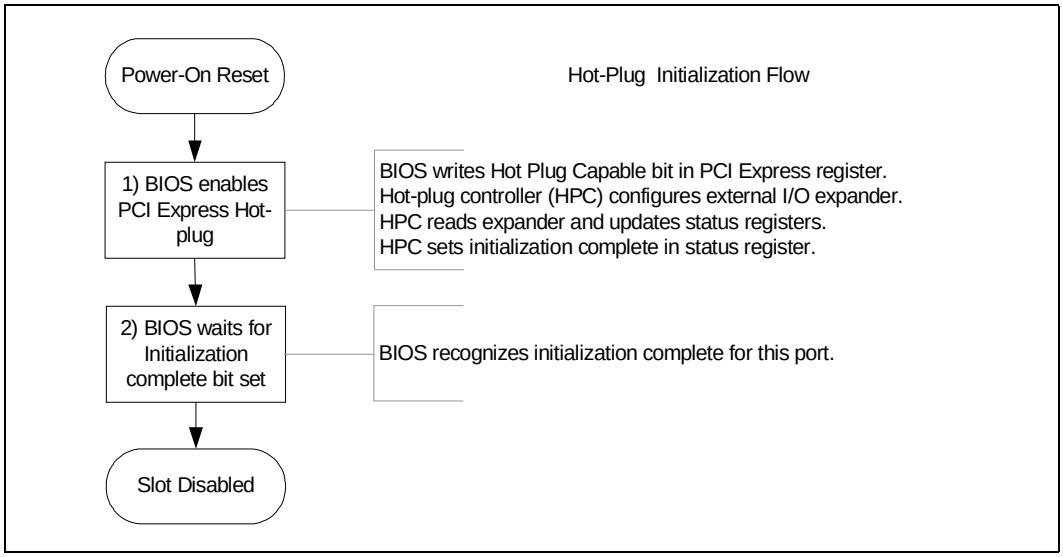


Figure 64. Hot-Plug Insertion Flow Via Pushbutton Request

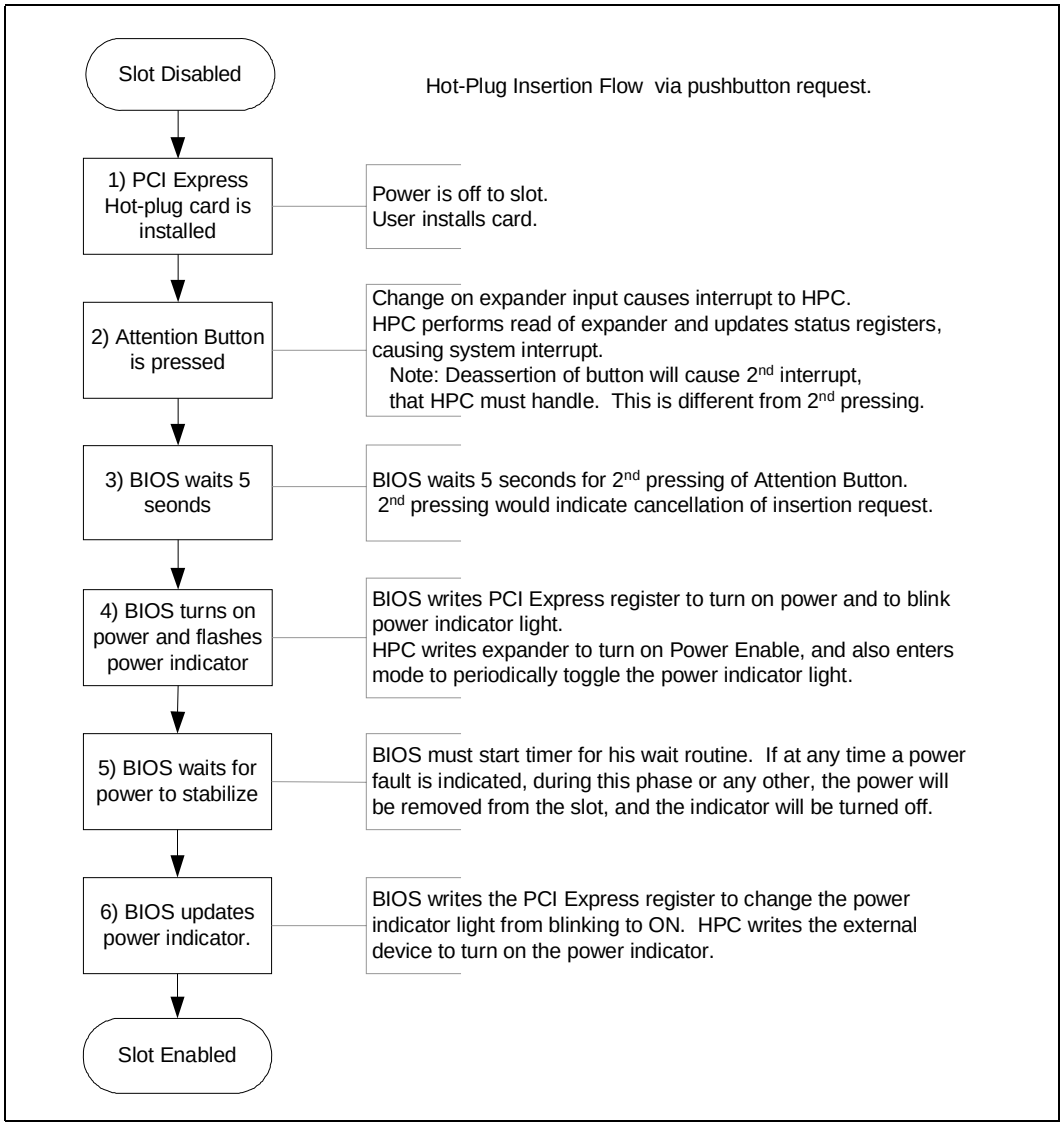
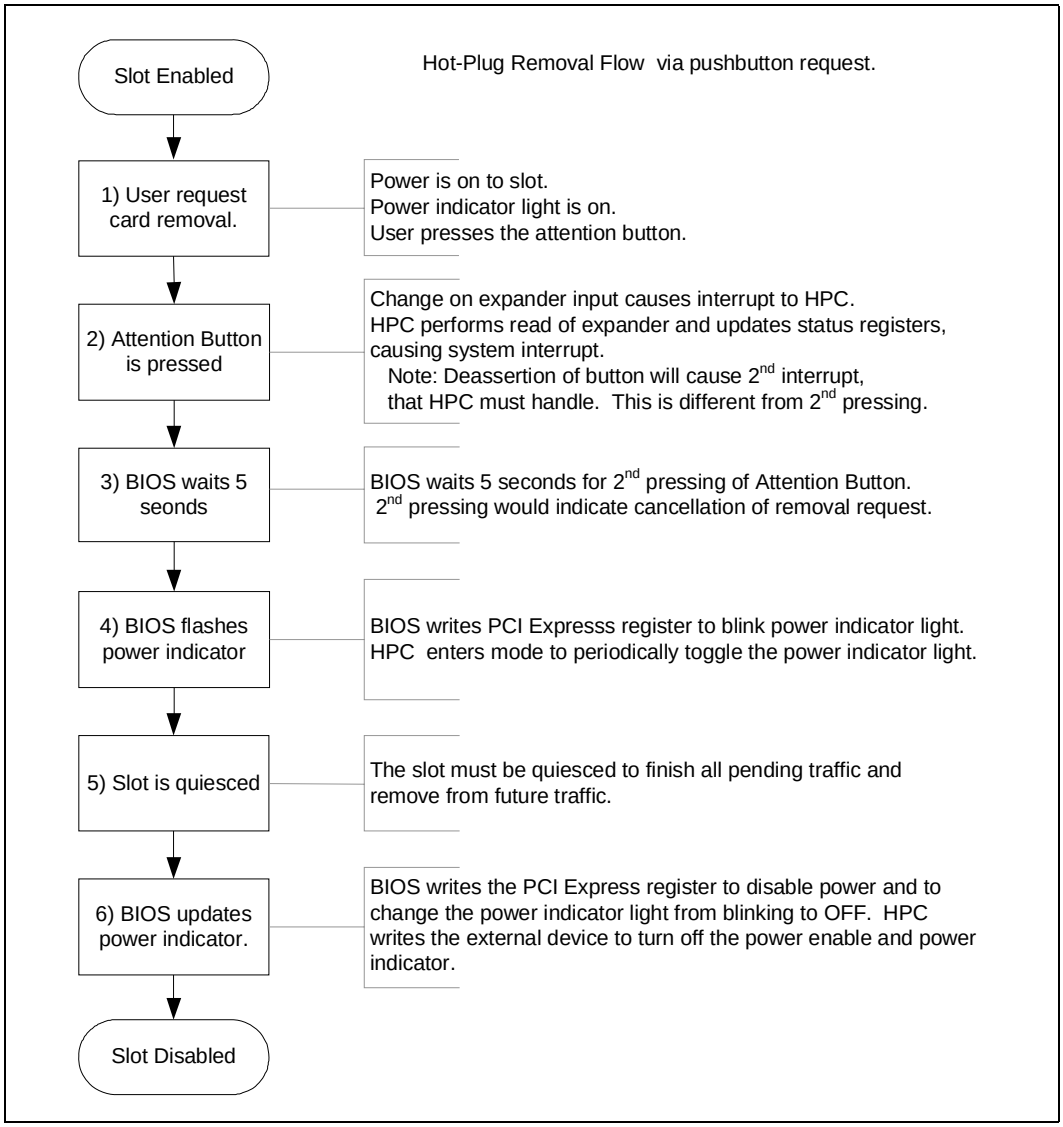


Figure 65. Hot-Plug Removal Flow Via Pushbutton



8.3.1.3.1 I/O Expander Features

The external I/O expander device provides the following capabilities:

- Configurable I/O
- Self power-up reset
- Interrupt output when inputs change
- Single-byte implementation
- SMB/I^2C interface (SMBus 2.0 compliant for arbitration)

8.3.1.4 I/O Expander SMB Packet Formats

As shown in the earlier diagram, the interface to the external I/O expander is the SMBus and an interrupt pin from the device when its inputs change. Figure 66 and Figure 67 show the SMBus packet format for a byte write and a read. The following symbols are defined: [S] is the start symbol, [A] is the ACK symbol, [W] indicates a write, [R] indicates a read, [NA] indicates a NAK, and [P] indicates the stop. Refer to the *SMBus Specification* for more details. The address for this device is composed of an assigned portion, and an incremental portion when multiple devices are used together. The command is essentially the register offset into the register stack within the external device. The data is either the data payload written or read back depending on the type of command.

Figure 66. SMB Byte Write Transaction for Expander

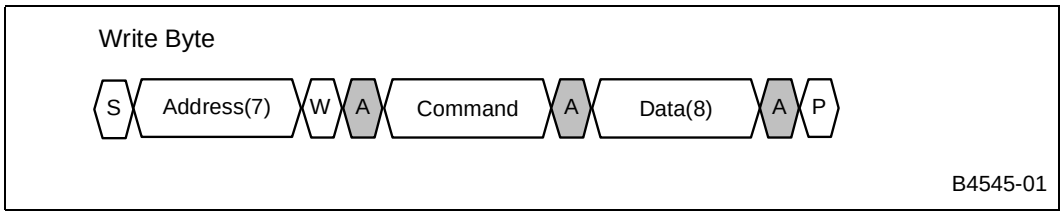
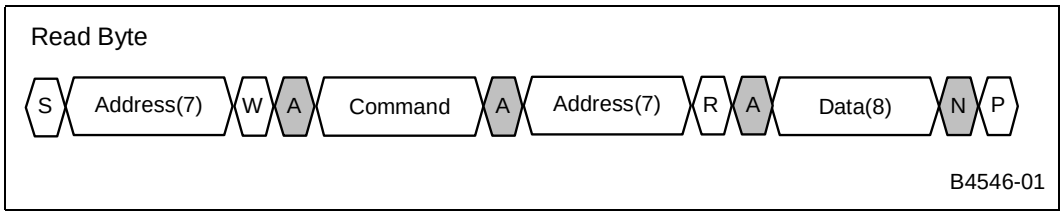


Figure 67. SMB Byte Read Transaction for Expander



8.3.1.5 I/O Expander Register Stack

The external I/O Expander has a particular register implementation that the hot-plug controller interfaces to directly. Table 119 defines the register stack for the single byte I/O expander. It has a single-port structure with four types of registers. Reading an input port reads the device pins for that port, whether the individual pins are configured as inputs or outputs. When reading the open-drain output port, you get what was last written to that port regardless of whether the pins associated with that port are configured as inputs or outputs. The output port defaults to all 1's. The polarity inversion register can be written to invert the polarity of the inputs read back; a '1' inverts, a '0' does not invert. The polarity register defaults to all zeros for no polarity inversion. Finally the configuration register defaults to all ones, which configures all the pins as input with high impedance output driver. The register bits must be written to a '0' to configure the pins as an output. The pins are tied to individual high value resistors pulling the pins to V_{CC} upon power up.

Table 119. Single Byte Register Stack

Command Byte	Register	Access	Default	Notes
0	Input Port	R	X	Reads the pin values
1	Output Port	R/W	FFh	Defaults to Open Drain Off
2	Polarity Inversion Port	R/W	00h	0 = No inversion of inputs (default) 1 = Inputs are inverted
3	Configuration Port	R/W	FFh	0 = Output 1 = Input (default)

8.3.1.6 I/O Expander Access/Event Matrix

Upon being enabled, the hot-plug controller configures the external device. A write to the PCI Express hot-plug registers in-turn causes an update of the corresponding register in the external device. Since the hot-plug controller has the blink function, depending on the setting of PCI Express register bits, it needs to update the external device at periodic intervals. When an input changes on the external device, the controller reads the external device to determine what changed. The matrix of these events and registers is shown below. The key for Table 120 is that e.g., W3 means write register 3, while R0 indicates read register 0.

Table 120. External Single-Byte I/O Expander Event Matrix

Hot-Plug Enable	Configuration	Update	Blink Timer Expired	EXP_HPIRQ# Asserted
0	Do Nothing	Do Nothing	Do Nothing	Do Nothing
1	Config (W3, W1, R0)	Update (W1)	Blink (W1)	Read In (R0)

8.3.1.7 IMCH Interface to External Device

The interface between the external expander device and the IMCH consists of the following pins:

- SCL serial clock line
- SDA serial data line
- Interrupt Output (open-drain)

8.3.1.8 Board Interface to External Device

The interface between the motherboard and the external expander device consists of the following pins:

- 3 Address input pins
- 8 Port I/O

8.3.1.9 I/O Expander Debug/Support Features

The following debug and support capabilities exist in the external expander architecture:

- Effect an interrupt from a configuration write
- Change upper address bits for external I/O expander device
- Last actual expander byte read available from configuration space

8.3.2 Initialization For Hot-Plug

To support “hot” addition of PCI Express devices while the system is up and running, the IMCH provides hooks for software resource allocation during initialization. Specifically, the IMCH is capable of exposing the logical PCI bus associated with a hot-plug ready PCI Express port even though that port is unpopulated and therefore non-functional at deassertion of reset. The exposed “bus” appears to software as the secondary side of the virtual PCI-to PCI bridge associated with the port in question. This allows the OS to allocate space for later use by the hot-plug ready port at such time as that interface becomes active. Internal status information for each PCI Express interface permits BIOS to differentiate between failed ports and those that are hot-plug

ready. Chapter 9.0, "System Clocking," includes a high level description of the sequence of events at initialization time resulting in a hot-plug ready state for one or more PCI Express interfaces.

Further hardware is dedicated to detecting an error condition and gracefully terminating (with master abort) any traffic directed to an PCI Express port in the hot-plug ready state. This includes configuration traffic, although such transactions are expected because enumeration software scans the logical bus for devices despite the fact that IMCH hardware "knows" the hot-plug ready port to be unpopulated. The IMCH provides independent master abort escalation controls to prevent reporting of configuration traffic to the inactive port as a system programming error.

8.3.3 Port Configuration Implications of Hot-Plug

During the boot process, the interaction of the BIOS and the IMCH PCI Express hardware determines the capability of the ports. The x8 port usage model is determined by the link training process. The PEA port can be utilized as one x8 port if all eight lanes train as a single link or can be used as two independent x4 links if both the 0 and 1 sides train as independent x4 links. If a non-hot-plug link fails to train at boot time, then that link is marked as NOT-PRESENT (NP) by BIOS. Hot-plug capable parts remain "PRESENT" whether they successfully train or not, as a connected device could be added after boot-time. Table 121 demonstrates the different personalities of a x8 dependent on hot-plug (HP), non-hot-plug, and devices present at boot-time (PAB). Note that only the 0 side of a native x8 link is hot-plug capable. Refer to Chapter 9.0, "System Clocking," for details on BIOS recommendations with respect to IMCH configuration after training.

Table 121. Native x8 Personalities After BOOT

Port PEA0			Port PEA1			Notes
Width	PAB	HP	Width	PAB	HP	
x8	Y	N	NP	N	N	Port PEA0 trained as x8
x4	Y	N	x4	Y	N	Port PEA0 trained as x4, Port PEA1 trained as x4
x4	Y	N	NP	N	N	Port PEA0 trained as x4
NP	N	N	x4	Y	N	Port PEA1 trained as x4
NP	N	N	NP	N	N	Neither Port trained
x8	Y	Y	NP	N	N	Port PEA0 trained as x8, available as x4
x4	Y	Y	x4	Y	N	Port PEA0 trained as x4, Port PEA1 trained as x4
x4	Y	Y	NP	N	N	Port PEA0 trained as x4, available as x8
x4	N	Y	x4	Y	N	Port PEA0 available as a x4 HP, PEA1 trained as x4
x8	N	Y	NP	N	N	Port PEA0 available as x8 or x4 HP

8.4 Platform Power Management Support

The IMCH is compatible with the *PCI Bus Power Management Interface Specification*, Revision 1.1 (referred to here as PCI-to-PMI). It is also compatible with the *Advanced Configuration and Power Interface Specification, Rev. 2.0 (ACPI)*. The Intel® 3100 Chipset is designed to operate seamlessly with operating systems employing these specifications.

The anticipated implementation for platform power management control is an add-on component connected to the IICH component of the core logic via its LPC and/or SMBus interfaces.

8.4.1 Supported System Power States

The IMCH and the system power states are analogous, thus no “device” power states are defined for the IMCH. As a result, the IMCH power state may be directly inferred from the system power state.

Like all systems, Intel® 3100 Chipset-based platforms must support the S0 (fully active) state at a minimum. The IMCH also supports S3 (suspend to RAM) and S5 (soft off).

In response to S3, the IMCH flushes all data from the internal coherent write buffer, sequences all active DIMM rows into their “self-refresh” state, and then return an Ack_Sx special cycle to the IICH. Upon completion of this sequence, the IMCH tolerates the removal of all clock references and power sources, save the DDR2 interface power. DDR2 interface power must be supplied so that the IMCH may hold the DIMMs in self-refresh. A full system initialization and configuration sequence is required upon system exit from the S3 state, as all (non-AUX) internal configuration information has been lost throughout the platform, but exit latency is much lower than it would be from S5, as the memory image has been maintained.

The extra internal logic support for S3 is not required for the S5 (soft off) system power state because all data in the memory array is lost regardless, and the coherent write buffer is architecturally part of the data stored in main memory.

8.4.1.0.1 Supported Processor Power States

Intel® 3100 Chipset based platforms support the C0 and C1 states as defined by the *Advanced Configuration and Power Interface Specification (ACPI)*. This implies that the core logic anchored by the IMCH properly understands and handles messaging between the IMCH and the FSB to facilitate transitions into and out of these states.

8.4.1.0.2 Supported Device Power States

The IMCH supports all PCI-to-PMI and PCI Express messaging required to place any subordinate device on any of its PCI Express ports into any of the defined device low power states. Peripherals attached to the PCI segments provided via a PXH component may be placed in any of their supported low power states via messaging directed from the IMCH through the intervening PCI Express hierarchy. Directly attached native PCI Express devices are not limited in their available low power states, although not all available states support the downstream device “wake-up” semantic.

Further detail on PCI Express power management support and accompanying PCI Express and subordinate device power management support are provided in Section 8.4.4, “PCI Express* Interface Power Management” on page 245.

8.4.1.0.3 Supported Bus Power States

No low power bus states are supported by the Intel® 3100 Chipset on its internal NSI interface between the IMCH and the IICH, nor does the IMCH support placement of the IICH only into any low-power state below D0, other than as a side-effect of placing the entire system into one of the S3 or S5 states.

Significant low power mode support is provided for the several IMCH PCI Express ports, as detailed in Section 8.4.4, “PCI Express* Interface Power Management” on page 245.

8.4.2 FSB Interface Power Management

No low power bus states are supported by the Intel® 3100 Chipset on its processor interface other than those achieved incidentally via placement of the processors themselves into sleep states.

8.4.3 DDR2 Interface Power Management

Hardware in the Intel® 3100 Chipset detects idle conditions on a per-chip-select basis on the DDR2 subsystem, and places the idle DIMM (or DIMM side) in a clock-disabled low-power state. When a new access decodes to a sleeping chip select, a single extra clock of latency is incurred to reenable the clock prior to issuing an activate cycle.

This power saving feature is above and beyond any software power management, and need only be enabled by system BIOS or firmware. No further software direction or interaction is required to realize the power savings from this feature.

No support is provided for self-refresh in the memory subsystem for this dynamic hardware-managed mechanism, thus the idled DRAM devices are dynamically awakened for refresh operations, and subsequently put back to sleep provided the idle condition persists. DDR2 self-refresh is supported as an integral piece of the S3 support.

8.4.4 PCI Express* Interface Power Management

In PCI Express, the traditional bus (B*) power states assigned to system buses are replaced by link (L*) power states, which are largely managed by hardware without software intervention. Entry into and out of these states may be initiated by two distinct mechanisms: traditional PCI-PMI type software managed state changes, and non-traditional PCI Express autonomous hardware state changes. The latter transition type is designated “Active State Power Management,” (ASPM) and with the *PCI Express Interface Specification, Rev. 1.0a*.

8.4.4.1 PCI Express* Link Power State Definitions

Support for all of the following PCI Express link power states is required for *PCI Express Specification* compatibility:

Note: The IMCH does not support all PCI Express link power states.

- L0 – Active state with all operations enabled (default state after platform initialization).
- L0s –**(Not supported)** Low latency, energy saving standby state, disabling exchange of both transaction layer packets and device link layer messages. This state is used exclusively by the ASPM PCI Express function, with entry and exit managed autonomously by PCI Express interface hardware. External PCI Express devices must never request entrance into L0s. If it does, undefined behavior will result. External device must set ASPM control register to “disable”, or 00b.
- L1 –**(Only supported in software managed state changes)** Moderate to high latency, very low power, standby state, disabling exchange of both transaction layer packets and device link layer messages. Entered when the downstream device is programmed to a device power state below the D0 active state, or optionally under hardware control during ASPM. The clock remains active in L1, and exit from this state may be initiated by either the upstream or the downstream device.
- L2/L3 Ready – Staging point for removal of main power and clocking. New intermediate state not directly related to PCI PM D-state transitions, nor to ASPM. Hand-shaking lands the link in this state in anticipation of power removal, at which point the link moves to either L2 or L3 depending upon the presence of Vaux.
- L2 – High latency, very low deep sleep state, disabling exchange of transaction layer packets and device link layer messages. L2 is characterized by removal of clocking and main power, but presence of Vaux power. Exit is initiated by restoring clocking and power, and full initialization.

- L3 – High latency, link off state with power, Vaux, and clock reference removed. Exit is initiated by restoring clocking and power, and full initialization.

The reader is referred to the *PCI Express* Interface Specification, Rev. 1.0a* for further detail on the link states and specific information on entry and exit mechanisms.

8.4.4.2 Software Controlled PCI Express* Link States

Software managed device power state changes do not explicitly control the power L-state of PCI Express links. Instead the L-state is inferred by hardware from the PCI-PMI power state of the devices attached to that link. When PM software transitions an PCI Express device to a low power state, that device automatically negotiates in hardware to bring its upstream link into the appropriate link power state.

No link is allowed to be in a link power state "below" that dictated by its attached components.

Table 122 defines the legal relationships between link and attached device power states.

Table 122. Relationship Between Link and Device PM States

Downstream Component D-State	Permissible Upstream Component D-State	Permissible Interconnect L-State
D0	D0	L0, L0s, L1†
D1	D0, D1	L1
D2	D0, D1, D2	L1
$D3_{hot}$	D0, D1, D2, $D3_{hot}$	L1, L2/L3 Ready
$D3_{cold}$	D0, D1, D2, $D3_{hot}$, $D3_{cold}$	L2, L3

† Entry into L0s or L1 while attached devices remain in D0 only occurs as a part of ASPM. Per the PCI Express spec, L0s support is mandatory, while L1 is optional. **L0s and L1 are not supported by ASPM**.

Several new semantics are introduced with PCI Express to support PCI-PMI compatible software managed device and link power state transitions. The majority of the new functionality is to accommodate an essentially edge-triggered in-band message scheme supporting multi chassis cabled system topologies, which must replace the function of traditional level-sensitive board traces for PM event and wake signaling. Further detail on PME signaling appears in Section 8.4.6, "PME Support" on page 248.

The IMCH supports messaging to facilitate transition of attached PCI Express devices to power states D0, D1, D2, and D3 (both $D3_{HOT}$ and $D3_{COLD}$). All attached devices are required by the *PCI Express Specification* to support the D0 and both D3 states, while D1 and D2 support are optional. It is up to software to confirm device support of the optional D1 and D2 states prior to attempting their use on any attached PCI Express device.

In the D1, D2, and $D3_{HOT}$ states the attached device is required to suppress initiation of any link traffic other than PME initiation (if enabled) as a master, and must only accept configuration transactions as a target. Functional context is maintained in the D1 and D2 states, such that full initialization of the attached device is not required upon the wake-up transition back to the D0 state. In both D3 states, functional context is not maintained, and full initialization is required after a transition back to D0.

Placing an attached device into a low power state results in automatic transition of the associated PCI Express port to its L1, L2 or L3 link state (depending upon the device power state). To save additional power in the L2 state, the platform power manager must remove the reference clock from the link. The IMCH does not provide the

necessary internal clock generation and distribution control to allow clock removal from one PCI Express port interface without impacting the operation of its peer ports on the IMCH. The Intel® 3100 Chipset does not provide support on its PCI Express link interfaces for the in-band "tone" required to wake from such a state.

8.4.4.3 Hardware Controlled PCI Express Link States

The IMCH does not support ASPM into the L0s or L1 states under hardware control. The Intel® 3100 Chipset negatively acknowledges (NAK) ASPM requests for L1 state transitions. IMCH configuration registers reflect this level of support for ASPM. External PCI Express devices must never request entrance into L0s. If it does, undefined behavior will result. External device must set ASPM control register to "disable", or 00b.

All ASPM functionality is disabled by default upon system power-up, and it is the responsibility of software to verify a viable platform clocking configuration prior to enabling ASPM functionality within the IMCH or in any attached PCI Express devices. In topologies where independent clock references are used at any point within the PCI Express subsystem hierarchy, the "fast training" sequence associated with ASPM is not guaranteed to successfully revive the associated link, and ASPM must remain disabled in the devices at both ends of that link.

8.4.4.4 System Clocking Solution Dependencies

The topology of the platform clocking solution dictates the viability of ASPM on each of the PCI Express links, because the nature of the clocks directly impacts the amount of time required to reacquire bit and symbol lock in the receiver after an arbitrarily long non-communicative period. When both ends of a link share a clock source, they "wander" together over the period they are out of communication with each other, and accordingly require a relatively brief period of training to reacquire lock. When the two ends of a link utilize completely independent clock references, they may become arbitrarily out of phase with each other while they are in low power states, and therefore require a significantly longer amount of time to reacquire lock upon waking. For this reason, the *PCI Express Interface Specification* provides for software discovery and communication of the actual clocking topology within the system prior to enabling the ASPM feature on any link within the system.

There are two primary components to the clocking discovery mechanism. First, all downstream ports, such as those on the IMCH root device, must report whether they use the same clock source as that provided to the slot (or down-device) connected to that port in the platform. This information is recorded in the Slot Clock Configuration bit of the Link Status Register for each port, and system BIOS is required to initialize these bits accordingly. Second, all add-in devices must report whether they utilize the clock reference provided on the add-in slot via the same bit in the same register of their capability structure.

System software may examine the settings of the Slot Clock Configuration bits of both the upstream and downstream devices for each port in the system, and determine whether a common clock reference is in use. This information is then communicated to both the upstream and the downstream devices via programming of the Common Clock Configuration bit of the Link Status Register. The setting of this bit determines the reported exit latency requirements for the L1 states. System software may then compare the exit latency requirements with the tolerated exit latencies of the attached device, and determine whether or not to enable ASPM for each link in the system. All ASPM functionality defaults to disabled at power-on, and remains so unless system software determines it may be enabled.

The "N_FTS" parameters exchanged during initial training corresponds to the "long" exit latencies associated with independent clocks, so if software later sets the Common Clock Configuration bits, it is also necessary to force link retraining in order to update the exchanged N_FTS information.

8.4.4.5 Device and Link PM Initialization

All PCI Express devices power-on into the $D0_{uninitialized}$ state, and remain in that non-communicative state until they have been configured and at least one of the Memory Space Enable, I/O Space Enable, or Bus Master Enable bits has been set by system software, at which point the device automatically transitions to the $D0_{ACTIVE}$ state indicative of normal operation.

8.4.5 Device and Slot Power Limits

All add-in devices must power-on to a state in which they limit their total power dissipation to a default maximum according to their form-factor. When BIOS updates the slot power limit register of the root ports within the IMCH, the IMCH automatically transmits a Set_Slot_Power_Limit message with corresponding information to the attached device. It is the responsibility of platform BIOS to properly configure the slot power limit registers in the IMCH, and failure to do so may result in attached endpoints remaining completely disabled in order to comply with the default power limitations associated with their form-factors.

8.4.6 PME Support

All information in this section refers to the IMCH PME support. See Chapter 22.0, "Power Management" for PME support in IICH.

In Intel® 3100 Chipset systems, only the system power manager or a device within the PCI Express hierarchy may initiate a power state change. Thus the only Power Management Event (PME) signaling support required in the IMCH is that associated with PCI Express. Note that a device bridging to another technology, such as a PXH bridging to PCI-X, may convert traditional PME signaling into PCI Express in-band PME messaging and thereby meet this requirement of the IMCH.

PME signaling in PCI Express is crafted to accomplish two distinct functions. First, it provides a signaling mechanism for devices requiring service to propagate a wake-up request to the power management controller. Secondly, it provides a messaging mechanism for devices requesting a power state change to pass their unique location within the PCI Express hierarchy to the power management controller. The combination of these two functions provides great flexibility and controllability for the power manager.

8.4.6.1 PME Wake Signaling

Wake signaling is only required to provide for device-initiated transition out of low power states where clock and/or power have been removed from the sleeping device. The PME mechanism does not require a wake-up function for attached devices still powered and receiving an interface reference clock, as devices in this state may simply initiate PME messaging directly. Wake is only required if the device wishing to initiate a PME message cannot do so without first requesting a change to the system clocking and power profile from the power management controller.

The wake signaling aspect of the system power management solution may vary in elegance and granularity. Depending upon the support level provided by the power management controller, a wake-up request from any given device may cause power and clocking to be restored to the entire system, only to the affected branch of the PCI Express hierarchy, or only to the requesting device.

While the *PCI Express Interface Specification* provides for two distinct wake signaling mechanisms, the Intel® 3100 Chipset supports only the legacy mechanism described below.

8.4.6.1.1 Legacy Wake Mechanism

The legacy wake signaling mechanism is directly analogous to that used in historical PCI-based system designs. In this case the platform architect is responsible for crafting paths routing collected wake signals between wake-capable devices and the management controller without participation from the IMCH and IICH equivalent devices. The collection of wake logic must run on auxiliary power, and must comprehend the potential for devices both with and without supplied auxiliary power co-existing on the same branch of the PCI Express hierarchy. Refer to the *PCI Express Interface Specification* for further details on legacy wake signaling.

While familiar and well understood, this mechanism does not provide for device-initiated wake-up in the fully A/C coupled implementation of a multi-chassis PCI Express based system solution. The limitation imposed upon the Intel® 3100 Chipset-based system is that remote-chassis devices must not be placed in a low-power state with the PCI Express link clocking and/or power disabled if legacy style wake signaling is desired for any peripheral in that remote chassis. It is still possible for software to place peripheral devices in low power sleep states, and to manage the device state of the PCI Express device attached to the inter-chassis cable. Devices in the remote chassis may still initiate power state changes via PME messaging provided the inter-chassis link has not been placed into an uncommunicative state. An alternative for the platform architect would be to place a compatible switch device between the IMCH and the remote chassis that supports the in-band wake mechanism described below, and rely on the switch to forward wake events to the management controller.

It is up to the platform architect to ensure that the power management controller can adequately isolate the source of a PME wake request as required to take appropriate power management
wake-up action.

8.4.6.1.2 In-Band PCI Express Wake Mechanism

PCI Express provides for an in-band mechanism whereby a device in a low power state attached via a link with clocking and/or power disabled may signal its desire to send a PME message using a “tone” on its otherwise uncommunicative link. Upstream devices must forward the wake-up request embodied in the “tone” to the root of the PCI Express hierarchy (IMCH), which must in turn forward the request to the system power management controller. This both eliminates the requirement for a side-band wake signal routed from the peripheral to the power manager, and provides an inter-chassis wake solution for multi-chassis system topologies.

Support for this “tone” mechanism is optional, and the IMCH does not offer this functionality.

8.4.6.2 PME Messaging

Once the link requesting a power state change has a communicative upstream link, it sends the PM_PME packet upstream towards the root device (IMCH), which in turn is responsible for notifying the management controller. This constitutes an in-band “virtual wire” signaling mechanism to replace the historical solution that involved multiple independent board traces routing PME requests to the power manager. Because the PM_PME propagates “in-band” on the PCI Express interface without any side-band signaling support, PME functionality is made available to multi-chassis system solutions.

The IMCH collects and “OR” PME requests from all logical PCI Express ports and propagate it to IICH over the NSI link as an Assert_PMEGPE message. The IICH then generates a specified interrupt to wake the power manager and invokes power management software. The interrupt service routine may then interrogate the various

PM status registers to determine the source(s) of PME. The IMCH would send Deassert_PMEGPE message over NSI link after power state change request has been serviced.

8.4.7 BIOS Support for PCI Express* PM Messaging

The *PCI Express Specification* stipulates hierarchical messaging semantics enforced by the root device (IMCH) to guarantee proper entry into and exit from unpowered device states. The ACPI BIOS must make special allowances for support of these semantics. There are two sets of messages that must be software-assisted to support power-off device states within the PCI Express hierarchy.

8.4.7.1 PCI Express* PME_TURN_OFF Semantic

Prior to removing power from any attached PCI Express links anywhere in the hierarchy, the root device must broadcast an PCI Express “PME_TURN_OFF” message to all downstream devices on the affected PCI Express port. The receiving devices propagate this message to all subordinate PCI Express ports (if any), collect “PME_TO_ACK” acknowledgement packets, and finally return a “PME_TO_ACK” transaction layer packet back to the root device. Once all active ports have acknowledged, the power management device may be notified that it is cleared to modify the collective power state of the PCI Express hierarchy. These message packets have posted semantics on the interface, thus the turn-off “pushes” all prior packets to their endpoints, and the acknowledge “pushes” any pending inbound traffic all the way to the root. This prevents “trapping” transactions or PME messages somewhere in the hierarchy at the time power is dropped, ultimately causing them to be lost.

In a pure PCI Express design, the PME_TURN_OFF packet would originate directly at the power manager, or perhaps at the IICH providing connection between the power manager and the remainder of the core logic. Neither the power manager nor the IICH is aware of the PCI Express messaging mechanism, thus the IMCH provides device-specific control and status bits for use by its ACPI BIOS.

The sequence of events to place a PCI Express device in an unpowered state is as follows:

- PCI-PM or ACPI compliant O/S software is called to place the system into a low-power sleep state (S3 or S5), prepares for suspension, and calls ACPI BIOS to carry out the platform power transition.
- The BIOS then communicates to the root complex that all PCI Express devices must prepare for power-off. This is accomplished through the device-specific configuration space of the internal virtual PCI-to-PCI bridges with subordinate PCI Express hierarchies. The BIOS must configure each active root port to power-down. When the configuration write is received to set the “PM Turn Off” bit, the associated root port transmits a PM_Turn_Off message downstream. At this point, any traffic in-flight continues to be handled normally by the IMCH – routed outbound, and completed inbound.
- The target PCI Express device ceases generation of new transactions inbound, waits for all pending transactions to complete, and prepares to lose power and clocking. If the target device has a subordinate hierarchy of its own, it propagates the PM_Turn_Off message downstream and wait for acknowledges from all subordinate ports. Once ready to be brought off-line, the target device issues a PM_TO_Ack TLP cycle in acknowledgement back to the root. Note that the link is still communicative at this point, with both power and clock available.
- After issuing the PM_TO_Ack cycle, the downstream device then issues a PM_Enter_L23 DLLP continuously upstream until it receives an acknowledge. In response to the PM_TO_Ack, the root port will commence the PCI Express handshake sequence necessary to set its “Turn Off Ack” status bit. In response to

the PM_Enter_L23 DLLP, the root transitions its downstream link to the electrical idle state. (This protocol sequence directly mirrors the L1 entry sequence.)

- ACPI BIOS, which has been waiting for all of the “Turn Off Ack” status bits to assert, now clears all the command and status bits associated with the PME_TURN_OFF. The routine then informs the power manager to go ahead with the change to the system power state.
  Note that software is required (by the PCI Express specification) to implement a “dead-man” timer such that a failure to receive a full complement of “Turn Off Ack” status bits does not result in an indefinite hang. This timeout is nominally 1 second, after which the power state change proceeds regardless.
- The power manager drops power and clocking to the target device(s), and all associated links automatically transition to either the L2 or L3 uncommunicative power states. The links enter L2 if Vaux is supplied by the platform, otherwise they enter L3. (Note that the IMCH does not support Vaux, so all downstream lanes will necessarily go to the L3 state.) The platform remains in the low-power state until a wake event is signaled.

In a fully PCI Express aware core logic implementation, the ACPI BIOS would not need to act as the interlock between the IMCH and the power manager, as all that functionality would be handled in hardware via direct messaging.

9.0 System Clocking

This chapter details the clocking requirements of the Intel® 3100 Chipset by stipulating supported operating frequency ranges, specifying all supported relative interface frequencies and the overclocking protection scheme. Operation outside the frequency domains stipulated here result in system instability and failures.

9.1 IMCH Clocking

Table 123 details IMCH clock interfaces, unidentified attributes and operational frequency ranges.

Table 123. Clocking Interfaces

Interface	Clock Type	Mnemonic	Fmin	Fmax	Comments
FSB (host)	Differential Input	HCLKIN	100 MHz[1]	167 MHz	Address at 2x, Data at 4x
DDR2 (mem)	Differential Output	DDRxCMDCLK	200 MHz	200 MHz	Geared to FSB
PCI Express*	Differential Input	PEA_CLK	100 MHz	100 MHz	SSC tolerant, transmit at 25x, asynchronous to FSB
TAP	Single-ended Input	TCLK	10 KHz	16.7 MHz	TAP clock, asynchronous to FSB
SMBus	Single-ended Input	SMBCLK	10 KHz	100 KHz	Asynchronous to FSB

Notes:
1. The FSB is designed to operate at 100, 133 and 167 MHz depending on the processor.

External clock synthesizers and/or distribution buffers are responsible for generating the differential host clock, express reference clock, NSI clock, SMBus clock, and the TAP clock (when utilized). All these clocks are phase-independent, and no mode is provided to recover the added latency incurred from crossing the resultant asynchronous boundary within the design (i.e., if system design steps are taken to reduce the number of oscillators and thereby provide phase-alignment on otherwise independent interfaces, no improvement in latency will result).

9.1.1 PCI Express* Clocking

The PCI Express clocking solution is in many ways analogous to that used by Infiniband (IBA) technology. The transceiver and directly associated hardware operate on a very high-speed clock derived by multiplying a reference by 25 (although IBA systems commonly use a multiple of 20), while the internal hardware that transfers information to and from those transceivers operates at one tenth of the transmit frequency. Also like IBA, the agents at either end of the bus may operate on references generated by completely independent oscillators with no specified phase relationship to each other, provided the frequencies are identical within tight tolerances. This clocking relationship has been designated “plesiochronous” within this document. In both PCI Express and

IBA, the receiver extracts the (remote) clock from the data bit stream, and utilizes specialized hardware to then move the resultant data back into the (local) transmit clock domain for transfer into the core of the device.

PCI Express differs from IBA in that Spread-Spectrum Clocking (SSC) is allowed in cases where the reference frequency is distributed across both ends of the port (rather than independent as just described). The reference frequencies supplied to all endpoints in the PCI Express subsystem must be generated from the same oscillator when SSC is desired, such that the frequency drift introduced is identical at both ends of every link. The IMCH introduces this limitation by sharing a single reference across all its PCI Express interfaces, thus requiring the platform designer to provide a board level solution that distributes clocks to all endpoints attached to the IMCH such that each reference meets the jitter requirements stipulated in the *PCI Express* Interface Specification, Rev. 1.0a*. Note that the "plesiochronous" nature of the interface provides a break on distribution skew, as the platform architect need not maintain any particular phase relationship between reference clocks delivered to the various PCI Express endpoints.

The Intel® 3100 Chipset supports a completely independent reference clock input for its PCI Express interfaces, allowing for either SSC or non-SSC platform solutions. This also allows the FSB and PCI Express frequencies to be varied independent of one another, which can be very useful in system debug situations. If any attached PCI Express device in a Intel® 3100 Chipset-based platform operates on its own local reference, then SSC must be disabled throughout the PCI Express subsystem.

9.1.2 DDR2 Geared Clocking

The Intel® 3100 Chipset generates, fans-out, and phase-aligns the DDR2 clock at the strapped gear ratios shown in Table 124 from the host clock reference. The ratio in use is under software control, and must be set properly via configuration register accesses during IMCH initialization (accessible from host, TAP, and SMBus alike). Note that an incorrect setting of the DDR2 gearing ratio results in unreliable memory operation.

The IMCH discovers its clocking environment in two distinct stages: at power-on, and much later via configuration directed by platform BIOS. At power-on the IMCH selects the divisor utilized by the core and DDR2 circuitry. It does this by reading the CPU_SEL[2:0]pins strapping.Refer to Table 124 for the required strapping or signal polarity for each supported FSB and DDR2 combination.

Table 124. CPU_SEL Divisor Selection

FSB Speed	DDR2 Speed	CPU_SEL[2:0]#	DDR2 Gearing Ratio
100 MHz, 400 MT/s	200 MHz, 400 MT/s DDR2-400	101	2:1
133 MHz, 533 MT/s	200 MHz, 400 MT/s DDR2-400	001	3:2
167 MHz, 667 MT/s	200 MHz, 400 MT/s DDR2-400	011	6:5

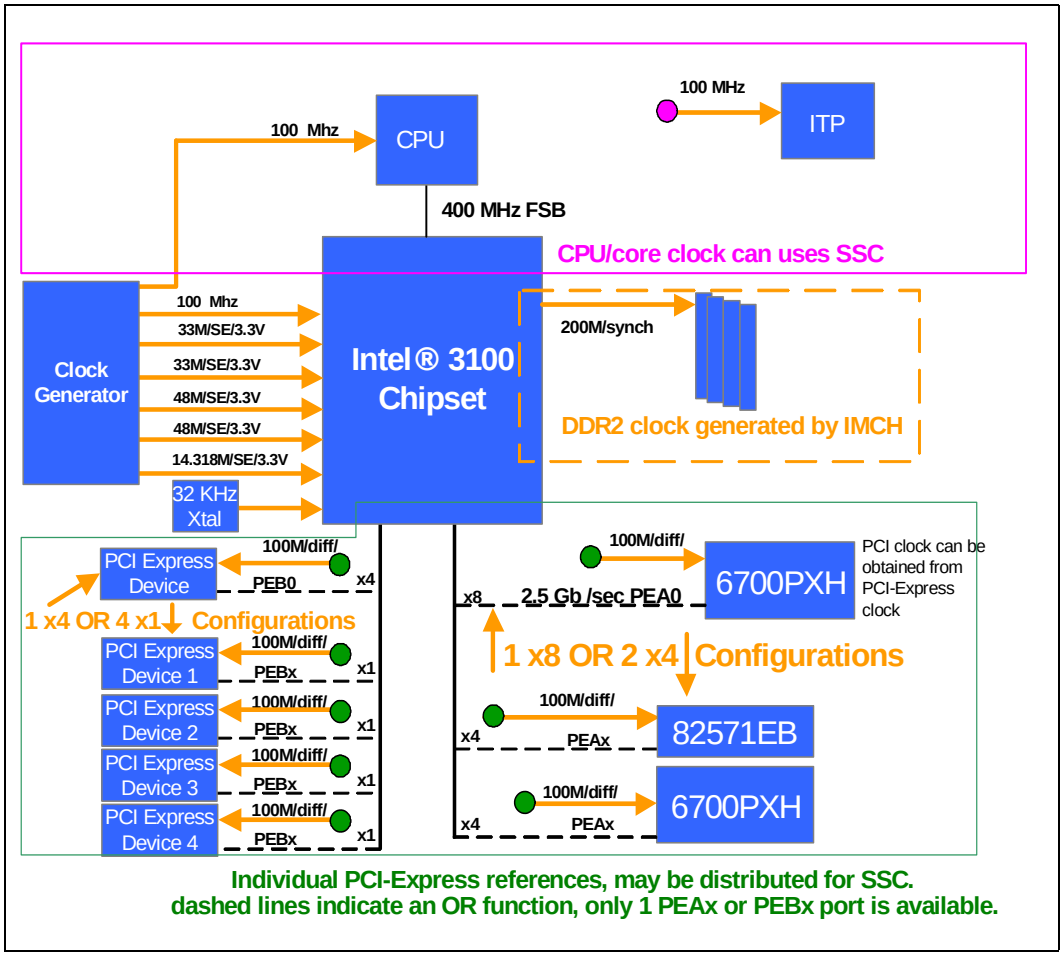
9.1.3 PCI Express* Port A (PEA) Platform Clocking Illustration

Figure 68 shows a representative system clocking solution for PCI Express port A (PEA). The example shown corresponds to a configuration with PEA configured to operate in either it's x8 mode or it's optional x4 mode.

A platform based on the Intel® 3100 Chipset has its choice of distributed or independent PCI Express clock reference solutions. This implementation is tolerant of FSB/core and PCI Express clock domains that are fully asynchronous to one another, thus both distributed and independent PCI Express clock reference solutions are supported.

Note: This figure is for illustration purposes only. Not all clocks may be shown. Please refer to the Intel® 3100 Chipset *Platform Design Guide* for complete details.

Figure 68. Platform Clocking



The platform architect has several options for how to generate and distribute the various clock references required by the platform.

9.1.4 Spread-Spectrum Clocking Limitations

A clock generation and distribution network is required if SSC is desired throughout the platform, including the PCI Express subsystem. The clock generator must produce 100, 133 and 167 MHz reference frequencies for the processor and the IMCH, as well as very low jitter 100 MHz reference frequencies for the IMCH and each of the PCI Express endpoint devices. Given a fully populated performance IMCH platform, this implies either a clock generation component with relatively high pin-count (over 100 pins), or a more modest generator with a independent low jitter differential clock buffer component.

The IMCH can tolerate a non-SSC PCI Express subsystem, where multiple endpoints receive independent and potentially differing reference frequencies. However, the IMCH cannot tolerate a mixed (SSC and non-SSC) PCI Express subsystem. Either all PCI Express endpoints utilize the same reference as the IMCH, and SSC may be enabled; or all PCI Express endpoints are assumed to be independent, and the entire PCI Express subsystem must operate without SSC. (It is strictly prohibited to create a configuration in which the two ends of any given PCI Express link receive independent SSC enabled reference clocks.)

9.2 IICH System Clocking

Table 125 lists the system clock domains. For complete details of the system clocking solution, refer to the system's clock generator component specification.

Table 125. IICH and System Clock Domains

Clock Domain	Frequency	Source	Usage
PCICLK	33.33 MHz	Main Clock Generator	Used for I/O devices and/or LPC Interface. This clock remains on during S0 state, and can be shut off during S3andS5. The PCICLK clock to peripherals may be shut off using the CLKRUN# protocol.
USB/SIO	48.00 MHz	Main Clock Generator	Used for external Super I/O and/or USB1 Controllers in the IICH. Shut off in S3 state.
OSC	14.31818 MHz	Main Clock Generator	Used for ACPI timer and Multimedia Timers. Shut off in S3 state.
RTC	32.768 kHz	IICH	RTC, Power Management. IICH has its own oscillator. Always running.
SATA Source Clock	100 MHz	Main Clock Generator	Differential clock pair used for SATA.
Source Clock	100 MHz	Main Clock Generator	Used by the PCI Express PLL to generate the clock domain at the PCI Express interface.
SATA Receive Domains	1.5 GHz, 150 MHz	SATA Devices	SATA Device-to-Host communication. Each Receive Port derives a 1.5 GHz clock from the differential receive pins; each port is independently derived. Shut off during S3 and S5.
SATA Transmit Domain	1.5 GHz, 150 MHz	IICH	SATA Host-to-Device communication.

Note: It is assumed that clocks start and stop cleanly (no glitches) based on STP_CPU#. Clocks are assumed to stop low, and have a clean first rising edge after starting.

10.0 System Reset

This chapter describes the reset signaling from the component and platform level.

10.1 Reset

The Intel® 3100 Chipset is the root of the I/O subsystem tree and is therefore responsible for general propagation of system reset throughout the platform. The Intel® 3100 Chipset must also facilitate any specialized synchronization of reset mechanisms required by the various system components. Details of the IMCH and IICH role in reset process are described in this chapter.

10.1.1 IMCH Reset Types

The IMCH differentiates among five types of reset as described in Table 126.

Table 126. IMCH Reset Classes

Type	Mechanism	Effect / Description
Power-Good	PWRGD input pin	Propagated throughout the system hierarchy. Resets all logic and state machines, and initializes *all registers* to their default states (sticky and non-sticky). Tri-states all IMCH outputs, or drives them to "safe" levels. Also known as Cold Reset.
Hard	Write to Reset Control register (I/O Offset CF9h)	Propagated throughout the system hierarchy. Resets all logic and state machines, and initializes *all non-sticky registers* to their default states. Tri-states all IMCH outputs, or drives them to "safe" levels. Also known as Hot Reset.
Processor-only	Configuration Write	Propagated to all processors via the CPURST# pin on the FSB. The IMCH does not undergo an internal reset.
Targeted	Configuration Write	Propagated down the targeted PCI Express* port hierarchy. Treated as a "Hard" reset by all affected components, clearing all state machines and non-sticky configuration registers.
BINIT#	Internal error handling propagated via FSB BINIT# pin.	Propagated to all FSB attached components (the IMCH and up to processors). Clears the IOQ, and resets all FSB arbiters and state machines to their default states. Not recoverable.

10.1.1.1 Power-Good Mechanism

The initial boot of a Intel® 3100 Chipset platform is facilitated by the Power-Good mechanism. The voltage sources from all platform power supplies are routed to a system component which tracks them as they ramp-up, asserting the platform "PWRGD" signal a fixed interval (nominally 2 ms) after the last voltage reference has stabilized. For specifications on supply ramp rates, refer to the Intel® 3100 Chipset *EDS Addendum*.

The IMCH and the IICH receives the system PWRGD signal via a dedicated pin as an asynchronous input, meaning that there is no assumed relationship between the assertion or deassertion of PWRGD and any system reference clock. When PWRGD is

deasserted all platform subsystems are held in their reset state. This is accomplished by various mechanisms on each of the different interfaces. The IMCH and the IICH holds in a power-on reset state when PWRGD is deasserted. The IICH also asserts its PLTRST# output and maintain its assertion for 1 ms after power is good.

The PCI Express attached devices and any hierarchy of components underneath them are held in reset via implicit messaging across the PCI Express interface. The IMCH is the root of the hierarchy, and does not engage in link training until power is good and the internal "hard" reset has deasserted. A PWRGD reset clears all internal state machines and logic, and initializes all registers to their default states, including "sticky" error status bits that are persistent through all other reset classes. To eliminate potential system reliability problems, all devices are also required to either tri-state their outputs or to drive them to "safe" levels during a power-on reset.

The only system information that "survives" a PWRGD reset is battery-backed or otherwise non-volatile storage (Flash, ROM, PROM, etc.).

10.1.1.2 Hard Reset Mechanism

Once the Intel® 3100 Chipset platform has been booted and configured, a full system reset may still be required to recover from system error conditions related to various device or subsystem failures. The "hard" reset mechanism is provided to accomplish this recovery without clearing the "sticky" error status bits useful to track down the cause of system reboot.

A hard reset is initiated by software through the Reset Control register (see Section 20.1.1.5) During a hard reset the Intel® 3100 Chipset drives PLTRST# active a minimum of 1 ms when initiated through the Reset Control register.

The Intel® 3100 Chipsetpropagates a hard reset to the FSB and to all subordinate PCI Express subsystems. The FSB components are reset via the CPURST# signal, while the PCI Express subsystems are reset implicitly when the root port links are taken down.

A hard reset clears all internal state machines and logic, and initializes all "non-sticky" registers to their default states. Note that although the error registers remain intact to facilitate root-cause of the hard reset, the Intel® 3100 Chipset platform in general requires a full configuration and initialization sequence to be brought back on-line (all other volatile configuration information is lost).

10.1.1.3 Processor-Only Reset Mechanism

For power management and other reasons, the Intel® 3100 Chipset supports a targeted processor only reset semantic. This mechanism was added to the platform architecture to eliminate double-reset to the system at large when reset-signaled processor information (such as clock gearing selection) must be updated during initialization bringing the system back to the S0 state after power had been removed from the processor complex.

10.1.1.4 Targeted Reset Mechanism

The targeted reset is provided for hot-plug events, as well as for port-specific error handling under MCA or SMI software control. The former usage model is new with PCI Express technology, and the reader is referred to the *PCI Express Interface Specification, Rev. 1.0a* for a description of the hot plug mechanism.

A targeted reset may be requested by setting bit six (Secondary Bus Reset) of the Bridge Control register (offset 3Eh) in the target root port device. This reset is identical to a general hard reset from the perspective of the destination PCI Express device; it does not differentiate at the next level down the hierarchy. Sticky error status survives in the destination device, but software is required to fully configure the port and all

attached devices once reset and error interrogation have completed. After clearing bit six, software may determine when the downstream targeted reset has effectively completed by monitoring the state of bit 1 (Link Active) of the VS_STS1 register (offset 47h) in the target root port device. This bit remains deasserted until the link has regained "link up" status, which implies that the downstream device has completed any internal and downstream resets, and successfully completed a full training sequence.

Under normal operating conditions it should not be necessary to initiate targeted resets to downstream devices, but the mechanism is provided to recover from combinations of fatal and uncorrectable errors which compromise continued link operation.

10.1.1.5 BINIT# Mechanism

The BINIT# mechanism is provided to facilitate CPU handling of system errors which result in a hang on the FSB. MCA code responding to an error indication (typically IERR# or MCERR#) attempts to interrogate the IMCH for error status, and if that FSB transaction fails to complete the processor automatically times out and responds by initiating a BINIT# sequence on the FSB.

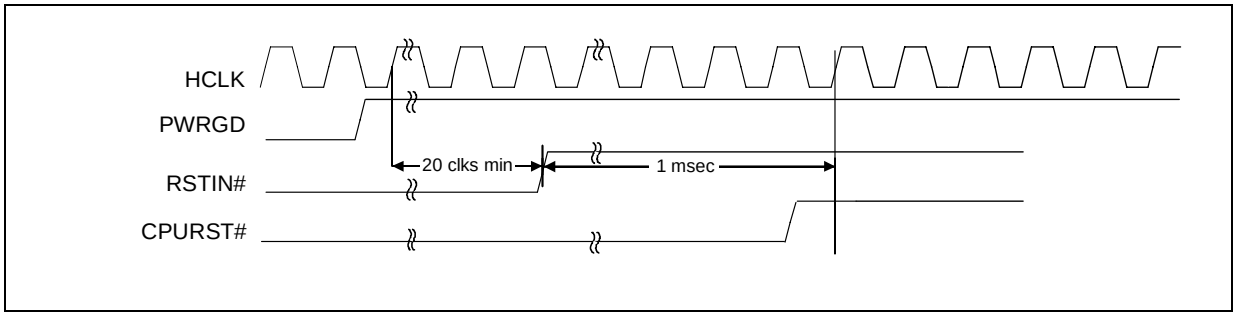
When BINIT# is asserted on the FSB, all agents (The Intel® 3100 Chipset's IMCH and all CPUs) are required to reset their internal FSB arbiters and all FSB tracking state machines and logic to their default states. This effectively "unhangs" the bus to provide a path into the Intel® 3100 Chipset configuration space. Note that the IMCH and PXH devices implement "sticky" error status bits, providing the platform software architect with free choice between BINIT# and a general hard reset to recover from a hung system.

Although BINIT# does not clear any configuration status from the system, it is not a recoverable event from which the platform may continue normal execution without first running a hard reset cycle. To guarantee that the FSB is cleared of any hang condition, the IMCH clears all pending transaction state within its internal traffic structures. This applies to outstanding FSB cycles as required, but also to in-flight memory transactions and inbound transactions. The resulting state of the platform is highly variable depending upon what precisely got wiped-out due to the BINIT# event, and it is not possible for hardware to guarantee that the resulting state of the machine supports continued operation. What the IMCH can guarantee is that no subordinate device has been reset due to this event (PCI Express links remain "up"), and that no internal configuration state (sticky or otherwise) has been lost. The IMCH also continues to maintain main memory via the refresh mechanism through a BINIT# event, thus machine-check software has access not only to the machine state, but also the memory state to track-down the source of the error.

10.1.2 Reset Sequencing

Figure 69 contains a timing diagram illustrating the progression through the power-on reset sequence. This is intended as a quick reference for system designers to clarify the requirements of the Intel® 3100 Chipset.

Figure 69. Power-On Reset Sequence



Note: The breaks in the HCLK waveform in Figure 69 are intended to illustrate further elapsed time in the interest of displaying a lengthy sequence in a single picture. Each of the delays in the reset sequence is of fixed duration, enforced by Intel® 3100 Chipset.

Note: HCLK in Figure 69 above is based on 167 MHz. All timings based on this clock will be different for other HCLK frequencies.

Table 127 summarizes the durations of the various reset stages illustrated in Figure 69, and attributes the delays to the component that enforces them.

The fixed delays (duration) in Table 127 provide time for subordinate PLL circuitry to lock on interfaces where the clock is withheld or resynchronized during the reset sequence.

Table 127. Reset Sequences and Durations

From	To	Duration	Source	Comment
Power on	PWRGD	>2 ms	Platform	Control logic on the platform must ensure that there are at least 2 ms of stable power before PWRGD is asserted.
PWRGD	PLTRST# deassertion	1 ms	Intel® 3100 Chipset	Intel® 3100 Chipset enforces delay between detecting PWRGD asserted and releasing PLTRST#
PLTRST# deassertion	Hard deassertion	2-8 HCLK	Intel® 3100 Chipset	Intel® 3100 Chipset waits for a common rising edge on all internal clocks, then releases core reset(s)
PLTRST# deassertion	CPURST# deassertion	1 ms	Intel® 3100 Chipset	Intel® 3100 Chipset enforces delay between PLTRST# and CPURST# deassertion.

10.1.2.1 Clocking Requirements at Power-On

The IMCH requires that the FSB reference clock is running and stable for a minimum of 2 ms prior to platform assertion of power-good. During this interval, the 1.5 V core power supply for the IMCH must also be stable, enabling the IMCH to lock its internal phase-locked loop circuitry.

Once power-good is asserted, the IMCH allows its PCI Express interfaces to train, while the IICH maintains assertion of PLTRST# for another 1 ms duration. This implies another system requirement; the PCI Express reference clock must also be stable for at least 1 ms prior to the assertion of PWRGD, allowing internal PLL circuitry to lock on both ends of each link before training is attempted. The training sequence propagates a hard reset throughout the system PCI Express hierarchy. The *PCI Express Interface Specification* stipulates a response time during training which guarantees that the reset has propagated in time to make training status available prior to the deassertion of

CPURST# on the FSB. This guarantee is necessary to ensure that BIOS code properly recognizes all logical PCI Express "root" ports in the IMCH prior to execution of the initial PCI bus enumeration sequence.

10.1.2.2 Configure Memory

The mandatory first step in memory configuration is frequency selection. The Intel® 3100 Chipset boots to the correct frequency based on the strapping of CPU_SEL[2:0] pins (see Table 70, "Clocks, Resets, and Miscellaneous Signals" on page 164). The straps must select the proper frequency depending on the base operating frequency of the FSB. The CPU_SEL[2:0] straps also select the appropriate gear ratio for DDR2 400 at the DDR2 interface. At reset, the Intel® 3100 Chipset disables the output clock reference drivers for all DIMM slots, which prevents them from locking at the wrong frequency, and then relocking after this step in the initialization process. It is assumed that the BIOS has *a priori* knowledge of the FSB frequency for the platform on which it is installed, and that DDR2 type is either also known or discovered via SPD access. If the target FSB and DDR2 frequencies differ from their default settings, BIOS must also update the gearing phase select registers (CLKGRFM0:Table 196 on page 336, CLKGRFM1:Table 197 on page 337, CLKGRMF0:Table 198 on page 337, CLKGRMF1:Table 199 on page 338) to guarantee reliable transfer of data between the core and DDR2 internal clock domains. Refer to the detailed descriptions of those registers (D0, F0, offsets A0-AFh) for required values at each of the supported gearing ratios. Once the gearing phase select registers have been updated, BIOS must configure the FSB frequency select field of the DRC register (bits 3:2), and the DRAM type field (bit 1) of that same register (D0, F0, offset 7Ch). Once the correct frequencies have been selected the appropriate gearing ratio updates automatically in the internal clock generation logic. BIOS must update CKDIS (D0, F0, offset 8Ch), such that only populated ranks of populated DIMM slots receive output clocks from the Intel® 3100 Chipset. BIOS must be aware of both the CKDIS operating mode and the CS routing topology in order to successfully complete this step. The motherboard clock routing topology for clock pins and for CS are assumed to be known to BIOS, either implicitly or via non-volatile configuration. The physical CS routing must be known to set up the DRM register (D0, F0, offset 80h) prior to moving on to DRA, DRB and the other "logical" DDR2 registers.

Internal hardware guarantees a "clean" output waveform as the FSB and DDR2 interface frequency selections are updated, by ensuring that there are no runt pulses or glitches on the internal or external clocks during the frequency transition. The internal design is such that the PLL within the Intel® 3100 Chipset does not require a relock period after the frequency update. BIOS must guarantee at least 200 µs of relock time for the PLL structures integrated into the registered DIMM devices prior to attempting communication across the external DDR2 interface. configuration may continue to take place during this interval.

10.2 IICH Reset

Information onIntel® 3100 Chipset's IICH reset mechanisms are described in detail in other sections of this document to reduce duplication of material and aid in keeping information synchronized. Section Section 10.2.1 IICH system Reset references will point to specific related references. The NSI reset sequence is described in detail in Section 10.2.2. For a detail description of power on and reset pin states refer to Chapter 4.0, "Power Requirements and Interface Signals."

10.2.1 IICH System Reset References

PWRBTN#: SYS_RESET#: PWROK: CPUPWRGD

10.2.2 NSI Reset Sequence

10.2.2.1 Processor Requirements

- CPURST# needs to be asserted at least for 1 ms and not more than 10 ms.
- The INIT strap from IICH to the processor needs to meet a hold time of 2 – 20 processor BCLKs with respect to deassertion of CPURST#.

10.2.2.2 Software Generated Processor-Only Reset

- The system is capable of resetting through software rather than a hardware mechanism.
- The reset bit resides in the IMCH. Once this bit is set, the IMCH asserts CPURESET# bin for 1 ms.
- No reset sequence message is sent to the IICH and the IMCH internal resets are kept deasserted.
- Once the 1 ms timer passes, the CPUREST# is deasserted.

11.0 Packaging

This chapter provides the ballout and package dimensions for the Intel® 3100 Chipset component. In addition, internal component package trace lengths to enable trace length compensation are listed.

The Intel® 3100 Chipset is packaged in a 1,284 ball, 40.0 x 40.0 mm, FC-BGA3 package.

The Intel® 3100 Chipset utilizes 1.092 mm bump pitch design to facilitate trace and impedance matching for the many high-speed interface signals that must escape the landing area.

The ball grid is 36x36, with three balls depopulated at each corner to allow for handling; this results in a total solder ball count of 1,284 on the landing side.

11.1 Package Specifications and Quadrant Layout

Figure 70 shows the package specifications. Figure 71 (Top View) and Figure 72 (Bottom View) show general quadrant (not exact component ball quadrant layout) information. Table 128 lists the ballout organized alphabetically by signal name. Table 129 lists the ballout with the listing organized numerically by ball number.

Figure 70. Package Specifications

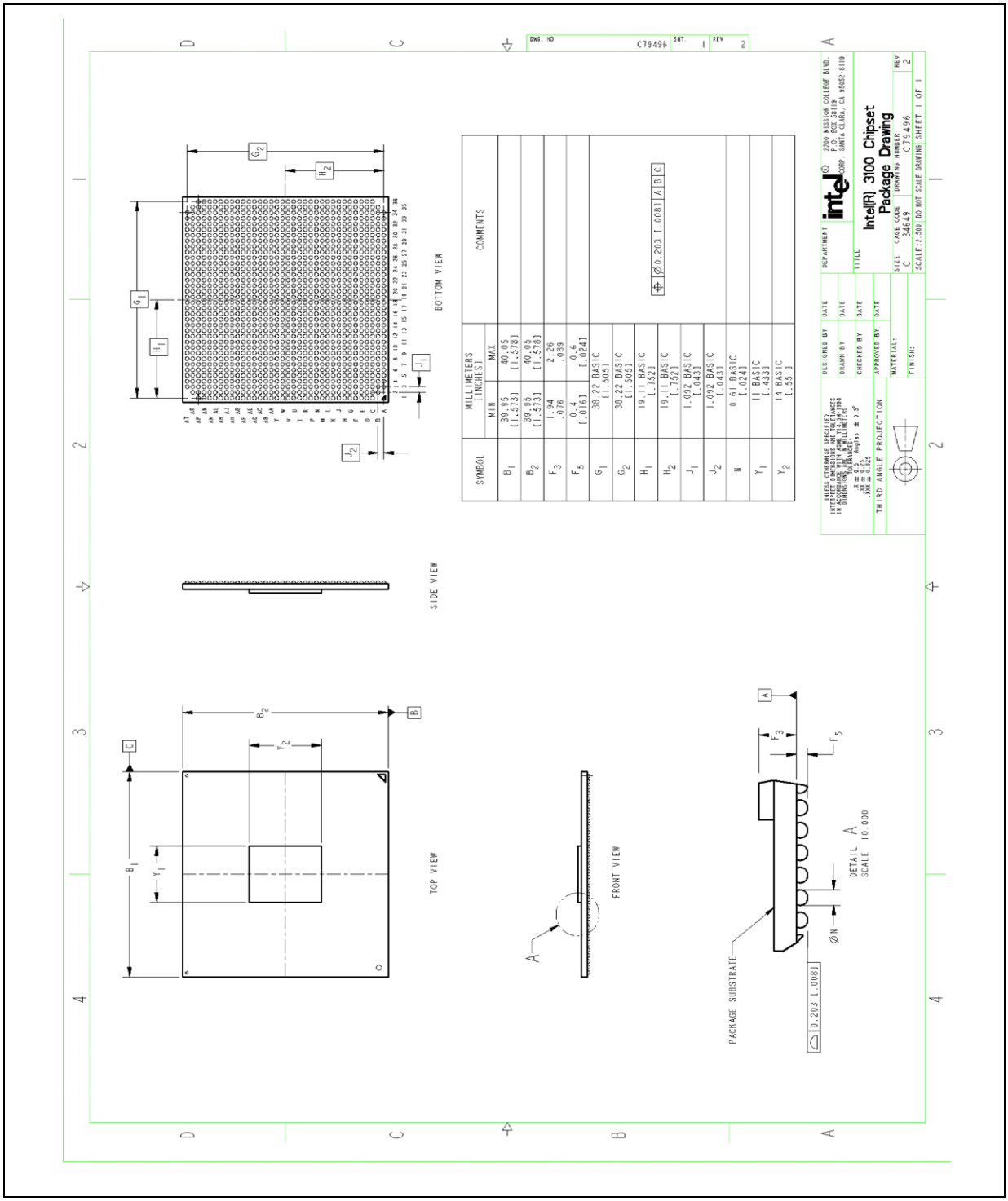


Figure 71. Intel® 3100 Chipset Quadrant Layout (Top View)

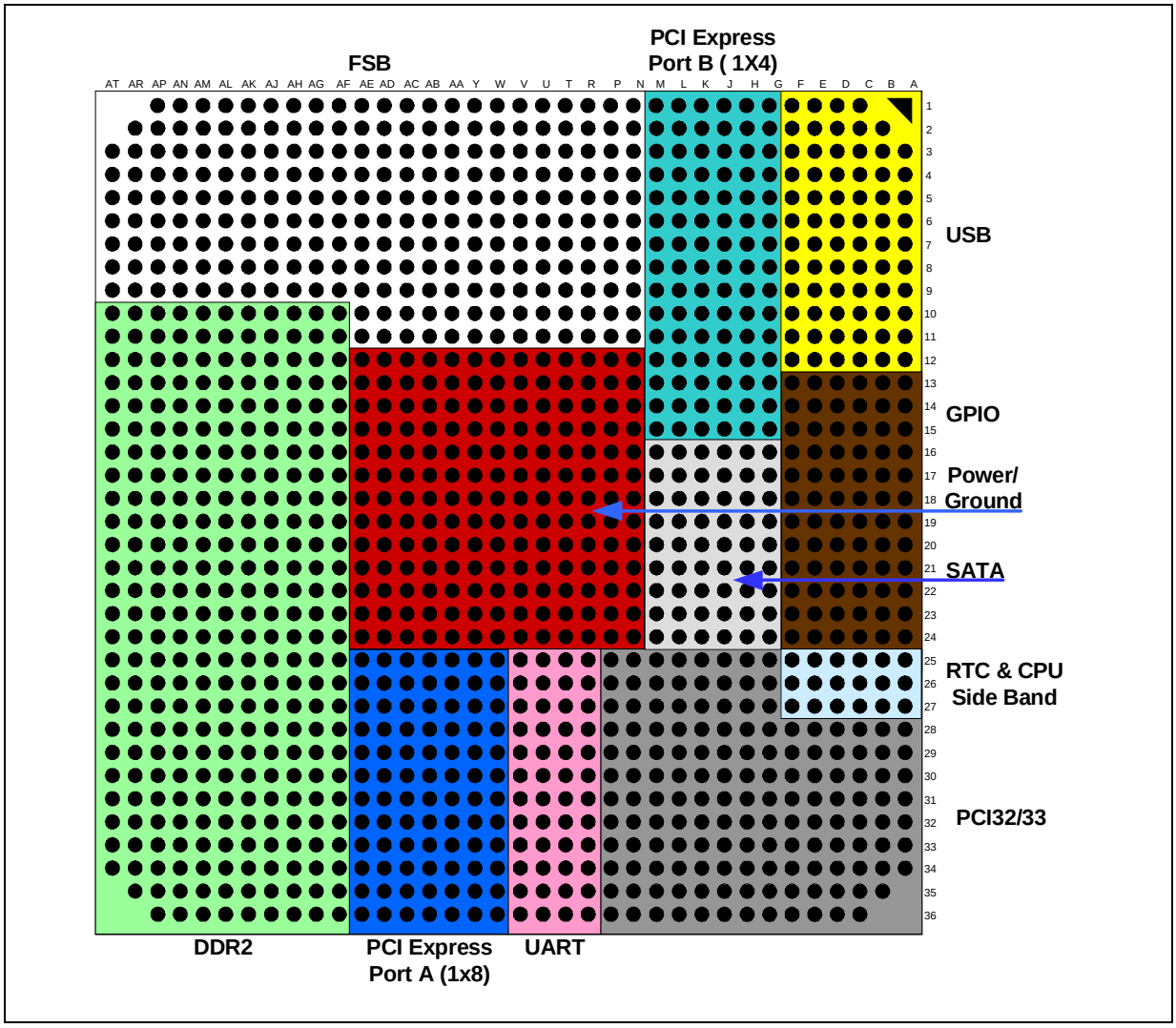
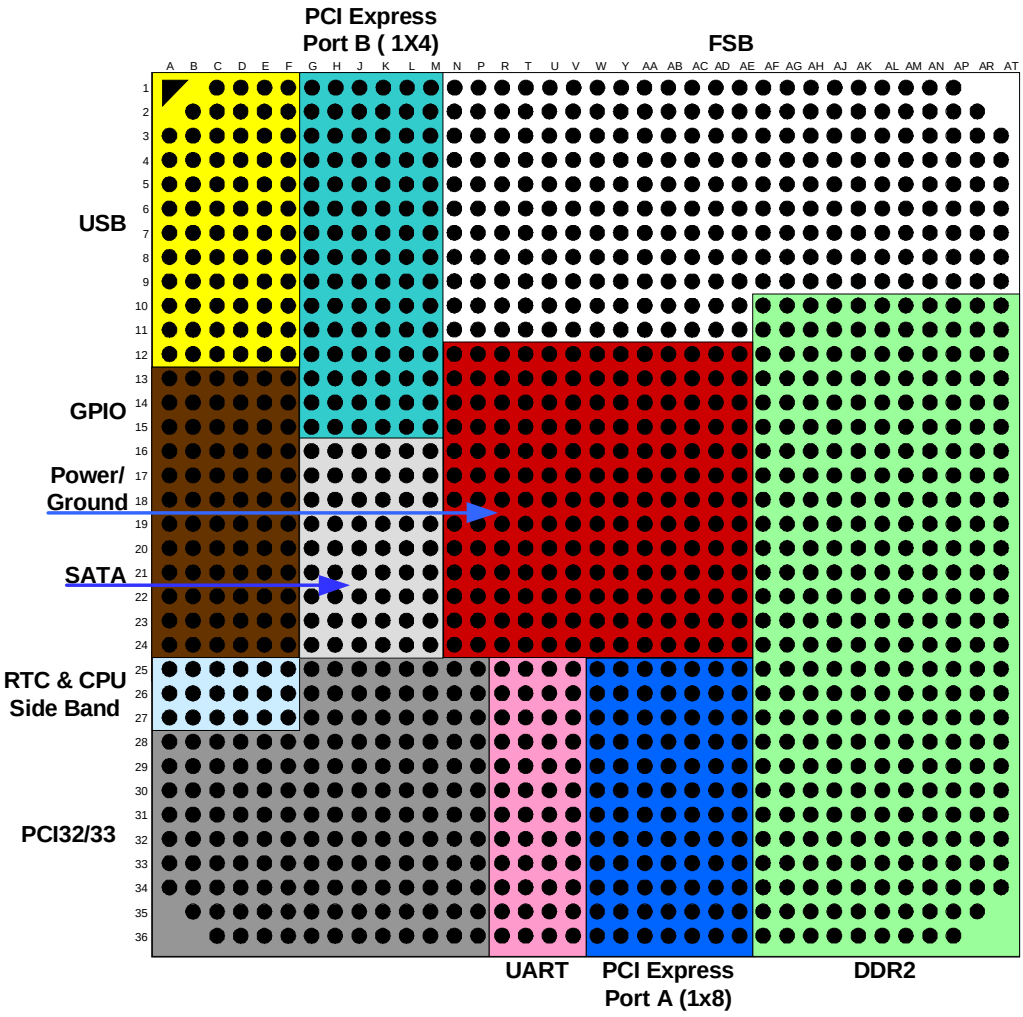


Figure 72. Intel® 3100 Chipset Quadrant Layout (Bottom View)



able 128. Signal Listing (Sheet 1 of 7)

Name	Ball #
ADS#	N3
AP[1]#	R8
AP[0]#	V10
BINIT#	R7
BNR#	R3
BPRI#	U10
BREQ[0]#	P4
BREQ[1]#	R1
CPURST#	P5
DBSY#	P2
DEFER#	T9
DP[3]#	V11
DP[2]#	U4
DP[1]#	R6
DP[0]#	R2
DINV[3]#	AP3
DINV[2]#	AK5
DINV[1]#	AF2
DINV[0]#	AG3
DRDY#	P3
HA[35]#	U2
HA[34]#	U1
HA[33]#	V1
HA[32]#	U3
HA[31]#	V3
HA[30]#	Y5
HA[29]#	V4
HA[28]#	AA2
HA[27]#	W3
HA[26]#	Y6
HA[25]#	AA6
HA[24]#	AA5
HA[23]#	Y2
HA[22]#	Y4
HA[21]#	Y1
HA[20]#	AA3
HA[19]#	AA1
HA[18]#	AA4
HA[17]#	V2
HA[16]#	W10
HA[15]#	AA10
HA[14]#	W9
HA[13]#	Y8
HA[12]#	V8
HA[11]#	U7
HA[10]#	V7
HA[9]#	AB8
HA[8]#	AA9
HA[7]#	U6
HA[6]#	AB9

Name	Ball #
HA[5]#	W8
HA[4]#	AA8
HA[3]#	AD9
HADSTB[1]#	Y3
HADSTB[0]#	U5
HACVREF	V9
HCLKINn	AH10
HCLKINp	AH11
HD[63]#	AR7
HD[62]#	AL8
HD[61]#	AP8
HD[60]#	AP7
HD[59]#	AT7
HD[58]#	AK8
HD[57]#	AL7
HD[56]#	AR8
HD[55]#	AM8
HD[54]#	AT5
HD[53]#	AR5
HD[52]#	AM5
HD[51]#	AN3
HD[50]#	AM6
HD[49]#	AP5
HD[48]#	AJ7
HD[47]#	AH8
HD[46]#	AK10
HD[45]#	AN2
HD[44]#	AN5
HD[43]#	AK7
HD[42]#	AK4
HD[41]#	AK3
HD[40]#	AK9
HD[39]#	AL1
HD[38]#	AJ2
HD[37]#	AM1
HD[36]#	AR4
HD[35]#	AM2
HD[34]#	AK2
HD[33]#	AH9
HD[32]#	AL9
HD[31]#	AF6
HD[30]#	AH3
HD[29]#	AD2
HD[28]#	AH5
HD[27]#	AH4
HD[26]#	AH1
HD[25]#	AF8
HD[24]#	AF5
HD[23]#	AD1
HD[22]#	AB1
HD[21]#	AB2

Name	Ball #
HD[20]#	AD3
HD[19]#	AH2
HD[18]#	AF4
HD[17]#	AC4
HD[16]#	AC3
HD[15]#	AF9
HD[14]#	AH7
HD[13]#	AD11
HD[12]#	AF7
HD[11]#	AG9
HD[10]#	AG6
HD[9]#	AE11
HD[8]#	AH6
HD[7]#	AC6
HD[6]#	AD8
HD[5]#	AE5
HD[4]#	AD7
HD[3]#	AF10
HD[2]#	AC5
HD[1]#	AD6
HD[0]#	AC7
HDSTBn[3]#	AP6
HDSTBn[2]#	AK6
HDSTBn[1]#	AE1
HDSTBn[0]#	AE4
HDSTBp[3]#	AN6
HDSTBp[2]#	AL6
HDSTBp[1]#	AE2
HDSTBp[0]#	AE3
HDVREF[1]	AJ10
HDVREF[0]	AD10
HIT#	U11
HITM#	T7
HLOCK#	T8
HREQ[4]#	W7
HREQ[3]#	AB10
HREQ[2]#	Y7
HREQ[1]#	AC10
HREQ[0]#	AA7
HTRDY#	P8
HCRES0	N6
HSLWCRES	P9
HODTCRES	T11
MCERR#	T5
RS[2]#	T6
RS[1]#	T4
RS[0]#	T3
RSP#	N5
DDR_MA[14]	AL33
DDR_MA[13]	AP10
DDR_MA[12]	AL34

Name	Ball #
DDR_MA[11]	AG25
DDR_MA[10]	AP21
DDR_MA[9]	AR34
DDR_MA[8]	AN28
DDR_MA[7]	AP32
DDR_MA[6]	AJ25
DDR_MA[5]	AL27
DDR_MA[4]	AR30
DDR_MA[3]	AJ23
DDR_MA[2]	AK23
DDR_MA[1]	AH22
DDR_MA[0]	AG21
DDR_BA[2]	AJ30
DDR_BA[1]	AJ20
DDR_BA[0]	AL20
DDR_DQ[63]	AJ14
DDR_DQ[62]	AL13
DDR_DQ[61]	AN15
DDR_DQ[60]	AN16
DDR_DQ[59]	AH14
DDR_DQ[58]	AR10
DDR_DQ[57]	AL14
DDR_DQ[56]	AG15
DDR_DQ[55]	AL12
DDR_DQ[54]	AT12
DDR_DQ[53]	AR15
DDR_DQ[52]	AR16
DDR_DQ[51]	AN11
DDR_DQ[50]	AK12
DDR_DQ[49]	AM14
DDR_DQ[48]	AT15
DDR_DQ[47]	AP17
DDR_DQ[46]	AR17
DDR_DQ[45]	AM19
DDR_DQ[44]	AN19
DDR_DQ[43]	AK16
DDR_DQ[42]	AK17
DDR_DQ[41]	AJ19
DDR_DQ[40]	AL19
DDR_DQ[39]	AL17
DDR_DQ[38]	AH17
DDR_DQ[37]	AT21
DDR_DQ[36]	AN21
DDR_DQ[35]	AG17
DDR_DQ[34]	AN17
DDR_DQ[33]	AR20
DDR_DQ[32]	AR21
DDR_DQ[31]	AT27
DDR_DQ[30]	AN27
DDR_DQ[29]	AP29
DDR_DQ[28]	AR31

Name	Ball #
DDR_DQ[27]	AT26
DDR_DQ[26]	AR26
DDR_DQ[25]	AP28
DDR_DQ[24]	AT30
DDR_DQ[23]	AG23
DDR_DQ[22]	AK24
DDR_DQ[21]	AL26
DDR_DQ[20]	AM27
DDR_DQ[19]	AG22
DDR_DQ[18]	AH23
DDR_DQ[17]	AK25
DDR_DQ[16]	AM26
DDR_DQ[15]	AR33
DDR_DQ[14]	AP33
DDR_DQ[13]	AM34
DDR_DQ[12]	AM33
DDR_DQ[11]	AT33
DDR_DQ[10]	AR32
DDR_DQ[9]	AN30
DDR_DQ[8]	AN29
DDR_DQ[7]	AM31
DDR_DQ[6]	AM30
DDR_DQ[5]	AH27
DDR_DQ[4]	AH26
DDR_DQ[3]	AN31
DDR_DQ[2]	AM32
DDR_DQ[1]	AK28
DDR_DQ[0]	AJ27
DDR_CB[7]	AL21
DDR_CB[6]	AK22
DDR_CB[5]	AP24
DDR_CB[4]	AM24
DDR_CB[3]	AH20
DDR_CB[2]	AK21
DDR_CB[1]	AM23
DDR_CB[0]	AR24
DDR_RAS#	AP20
DDR_CAS#	AM17
DDR_WE#	AP19
DDR_CS[7]#	AH12
DDR_CS[6]#	AJ15
DDR_CS[5]#	AL11
DDR_CS[4]#	AM16
DDR_CS[3]#	AM15
DDR_CS[2]#	AJ16
DDR_CS[1]#	AK15
DDR_CS[0]#	AG16
DDR_CMDCLKp [3]	AR23
DDR_CMDCLKp [2]	AR25
DDR_CMDCLKp [1]	AJ21

Name	Ball #
DDR_CMDCLKp [0]	AT22
DDR_CMDCLKn [3]	AR22
DDR_CMDCLKn [2]	AP25
DDR_CMDCLKn [1]	AH21
DDR_CMDCLKn [0]	AT23
DDR_DQSp[17]	AN23
DDR_DQSp[16]	AR13
DDR_DQSp[15]	AR14
DDR_DQSp[14]	AR18
DDR_DQSp[13]	AG18
DDR_DQSp[12]	AT29
DDR_DQSp[11]	AP26
DDR_DQSp[10]	AN35
DDR_DQSp[9]	AK29
DDR_DQSp[8]	AL22
DDR_DQSp[7]	AT11
DDR_DQSp[6]	AR12
DDR_DQSp[5]	AM18
DDR_DQSp[4]	AL18
DDR_DQSp[3]	AP27
DDR_DQSp[2]	AL25
DDR_DQSp[1]	AP34
DDR_DQSp[0]	AL30
DDR_DQSn[17]	AN22
DDR_DQSn[16]	AP13
DDR_DQSn[15]	AP14
DDR_DQSn[14]	AT18
DDR_DQSn[13]	AH18
DDR_DQSn[12]	AR29
DDR_DQSn[11]	AN26
DDR_DQSn[10]	AN34
DDR_DQSn[9]	AK30
DDR_DQSn[8]	AM22
DDR_DQSn[7]	AR11
DDR_DQSn[6]	AN12
DDR_DQSn[5]	AN18
DDR_DQSn[4]	AK18
DDR_DQSn[3]	AR28

Name	Ball #
DDR_DQSn[2]	AM25
DDR_DQSn[1]	AP35
DDR_DQSn[0]	AL29
DDR_VREF	AG24
DDR_CKE[3]	AK35
DDR_CKE[2]	AJ26
DDR_CKE[1]	AL28
DDR_CKE[0]	AK32
DDR_CRES0	AJ36
DDR_IMPCRES	AH28
DDR_RES1	AK36
DDR_RES2	AH29
DDR_SLWCRES	AK31
PEA0_Tn[7]	AC29
PEA0_Tn[6]	AE29
PEA0_Tn[5]	AF31
PEA0_Tn[4]	AE35
PEA0_Tn[3]	AA29
PEA0_Tn[2]	Y35
PEA0_Tn[1]	AB34
PEA0_Tn[0]	AA33
PEA0_Tp[7]	AC30
PEA0_Tp[6]	AE30
PEA0_Tp[5]	AF32
PEA0_Tp[4]	AE36
PEA0_Tp[3]	AA30
PEA0_Tp[2]	Y36
PEA0_Tp[1]	AB35
PEA0_Tp[0]	AA34
PEA0_Rn[7]	AE32
PEA0_Rn[6]	AD28
PEA0_Rn[5]	AF34
PEA0_Rn[4]	AD33
PEA0_Rn[3]	AA26
PEA0_Rn[2]	Y32
PEA0_Rn[1]	AB30
PEA0_Rn[0]	Y28
PEA0_Rp[7]	AE33
PEA0_Rp[6]	AD29
PEA0_Rp[5]	AF35
PEA0_Rp[4]	AD34
PEA0_Rp[3]	AA27
PEA0_Rp[2]	Y33
PEA0_Rp[1]	AB31
PEA0_Rp[0]	Y29
PEA_ICOMPI	AC36
PEA_RCOMPO	AC35
PEA_CLKp	AC33
PEA_CLKn	AC32
PEB0_Tp[3]	L4
PEB0_Tp[2]	K3
PEB0_Tp[1]	H2
PEB0_Tp[0]	M6

Name	Ball #
PEB0_Tn[3]	L5
PEB0_Tn[2]	K2
PEB0_Tn[1]	H1
PEB0_Tn[0]	M7
PEB0_Rp[3]	M3
PEB0_Rp[2]	K5
PEB0_Rp[1]	J4
PEB0_Rp[0]	L7
PEB0_Rn[3]	M4
PEB0_Rn[2]	K6
PEB0_Rn[1]	J3
PEB0_Rn[0]	L8
PEB_CLKp	M1
PEB_CLKn	L1
PEB_ICOMPI	L2
PEB_RCOMPO	J1
PEB_RPC[0]	T33
PEB_RPC[1]	R31
SMBSCL	AJ35
SMBSDA	AJ32
SMBALERT#\GPI[11]	E26
SMBDATA	H29
SMBCLK	G30
INTRUDER#	C28
SMLINK[1]	F30
SMLINK[0]	D30
PWRGD	AG29
RSTIN#	AG30
PE_HPINTR#	AG28
CPU_SEL[2]	AL36
CPU_SEL[1]	AF26
CPU_SEL[0]	AL35
CLK14	B34
CLK48	H11
SPKR	C33
RTEST#	A29
PCICLK	M28
PLTRST#	P32
INTVRMEN	C25
GPI[0]	A33
GPI[1]	D35
GPI[6]	G17
GPI[7]	E17
GPI[8]	G27
GPI[9]	E4
GPI[10]	F1
GPI[12]	G18
GPI[13]	A18
GPI[14]	F6
GPI[15]	F7
GPO[16]	D36
GPO[17]	A32
GPO[18]	B17

Name	Ball #
GPO[19]	A17
GPO[20]	B16
GPO[21]	C16
GPO[23]	C15
GPIO[24]	D26
GPIO[25]	F27
GPI[26]	D18
GPIO[27]	B28
GPIO[28]	J28
GPI[29]	F18
GPI[30]	C18
GPI[31]	E19
GPIO[32]	F16
GPIO[33]	E16
GPIO[34]	D17
GPI[40]	K30
GPO[48]	P34
TCK	AG36
TRST#	AG32
TMS	AF28
TDI	AJ34
TDO	AH33
TEST#	AE28
DEBUG[7]	AG34
DEBUG[6]	AF29
DEBUG[5]	AH30
DEBUG[4]	AF27
DEBUG[3]	AG33
DEBUG[2]	AE27
DEBUG[1]	AG35
DEBUG[0]	AG31
WL_PU[1]	E28
WL_PU[0]	F19
USBp[3]	C7
USBp[2]	E8
USBp[1]	D9
USBp[0]	C10
USBn[3]	C6
USBn[2]	E7
USBn[1]	D8
USBn[0]	C9
USB_RBIASp	J12
USB_RBIASn	J13
OC[3]#	E1
OC[2]#	E2
OC[1]#	D6
OC[0]#	D5
SATA_CLKp	G24
SATA_CLKn	G23
SATA_RBIASp	J25
SATA_RBIASn	H25
SATA_LED#	K23
SATA_TXp[5]	L19

Name	Ball #
SATA_TXp[4]	J21
SATA_TXp[3]	H19
SATA_TXp[2]	L16
SATA_TXp[1]	J15
SATA_TXp[0]	H13
SATA_TXn[5]	L20
SATA_TXn[4]	J22
SATA_TXn[3]	H20
SATA_TXn[2]	L17
SATA_TXn[1]	J16
SATA_TXn[0]	H14
SATA_RXp[5]	H23
SATA_RXp[4]	K21
SATA_RXp[3]	G21
SATA_RXp[2]	K18
SATA_RXp[1]	H17
SATA_RXp[0]	K15
SATA_RXn[5]	H22
SATA_RXn[4]	K20
SATA_RXn[3]	G20
SATA_RXn[2]	K17
SATA_RXn[1]	H16
SATA_RXn[0]	K14
RTCX1	B26
RTCX2	A27
WDT_TOUT#	W27
SERIRQ	J24
PIRQ[A]#	C31
PIRQ[B]#	A30
PIRQ[C]#	B29
PIRQ[D]#	C30
PIRQ[E]#/ GPI[2]	D32
PIRQ[F]#/ GPI[3]	E31
PIRQ[G]#/ GPI[4]	F31
PIRQ[H]#/ GPI[5]	E32
A20GATE	E22
A20M#	A21
FERR#	A24
IGNNE#	A23
INIT#	D23
INIT3_3V#	E23
CPUPWRGD/ GPO[49]	F22
INTR	C21
CPUSLP#	D21
NMI	B22
STPCLK#	C22
SMI#	B23
RCIN#	C24
RI#	J27
SLP_S3#	D29

Name	Ball #
SLP_S4#	E29
SLP_S5#	G29
SUS_STAT#/ LPCPD#	H28
SUSCLK	F25
THRM#	F24
THRMTRIP#	B32
VRMPWRGD/ VGATE	K22
PWROK	B25
PWRBTN#	F28
RSMRST#	A26
SYS_RESET#	K24
WAKE#	E25
PME#	P31
LAD[3]/ FWH[3]	C19
LAD[2]/ FWH[2]	B19
LAD[1]/ FWH[1]	B20
LAD[0]/ FWH[0]	A20
LFRAME#/ FWH[4]	D20
LDRQ[1]#/ GPI[41]	F21
LDRQ[0]#	E20
UART_CLK	W30
SIU_RXD[2]	V30
SIU_RXD[1]	V27
SIU_TXD[2]	V33
SIU_TXD[1]	V35
SIU_CTS[2]#	W34
SIU_CTS[1]#	V36
SIU_DSR[2]#	W33
SIU_DSR[1]#	W35
SIU_DCD[2]#	V32
SIU_DCD[1]#	V34
SIU_RI[2]#	W28
SIU_RI[1]#	V29
SIU_DTR[2]#	W29
SIU_DTR[1]#	V26
SIU_RTS[2]#	W32
SIU_RTS[1]#	V31
AD[31]	G32
AD[30]	F33
AD[29]	F34
AD[28]	H32
AD[27]	G33
AD[26]	J31
AD[25]	J30
AD[24]	H34
AD[23]	L32
AD[22]	L31

Name	Ball #
AD[21]	R34
AD[20]	M30
AD[19]	M31
AD[18]	U35
AD[17]	R30
AD[16]	T30
AD[15]	K36
AD[14]	J34
AD[13]	K32
AD[12]	K33
AD[11]	M36
AD[10]	L34
AD[9]	N35
AD[8]	N36
AD[7]	L29
AD[6]	M34
AD[5]	M33
AD[4]	N32
AD[3]	N33
AD[2]	R36
AD[1]	T35
AD[0]	T36
C/BE[3]#	J33
C/BE[2]#	N30
C/BE[1]#	L35
C/BE[0]#	K29
DEVSEL#	F36
FRAME#	T29
IRDY#	N29
TRDY#	P28
STOP#	G35
PAR	K35
PERR#	G36
REQ[1]#	E34
REQ[0]#	H31
GNT[1]#	P29
GNT[0]#	D33
PLOCK#	H35
SERR#	J36
PCIRST#	C34
V5REF	U29
V5REF	D14
V5REFSUS	E5
VCC_CORE_PLL	AG12
VCC_DDR_PLL	AF11
VCC_CPU	C23
VCC_CPU	D24
VCC_CPU	B24
VCC_DDR	AE18
VCC_DDR	AE20
VCC_DDR	AE22
VCC_DDR	AE24
VCC_DDR	AE26

Name	Ball #
VCC_DDR	AM12
VCC_DDR	AL16
VCC_DDR	AK14
VCC_DDR	AJ12
VCC_DDR	AH16
VCC_DDR	AF15
VCC_DDR	AF13
VCC_DDR	AM21
VCC_DDR	AK20
VCC_DDR	AJ22
VCC_DDR	AJ18
VCC_DDR	AG20
VCC_DDR	AF23
VCC_DDR	AF21
VCC_DDR	AF19
VCC_DDR	AF17
VCC_DDR	AJ24
VCC_DDR	AM29
VCC_DDR	AL24
VCC_DDR	AK27
VCC_DDR	AJ29
VCC_DDR	AH25
VCC_DDR	AN33
VCC_DDR	AF25
VCC_DDR	AM36
VCC_DDR	AL31
VCC_DDR	AK34
VCC_DDR	AH36
VCC_DDR	AH32
VCC_DDR	AP12
VCC_DDR	AT16
VCC_DDR	AT14
VCC_DDR	AP16
VCC_DDR	AN14
VCC_DDR	AN20
VCC_DDR	AT20
VCC_DDR	AP23
VCC_DDR	AT28
VCC_DDR	AT25
VCC_DDR	AT32
VCC_DDR	AP31
VCC1_5	V13
VCC1_5	T13
VCC1_5	P13
VCC1_5	N14
VCC1_5	N16
VCC1_5	P15
VCC1_5	R16
VCC1_5	T15
VCC1_5	U14
VCC1_5	K27
VCC1_5	F20
VCC1_5	F17

Name	Ball #
VCC1_5	M18
VCC1_5	M2
VCC1_5	N7
VCC1_5	N4
VCC1_5	P12
VCC1_5	P11
VCC1_5	P10
VCC1_5	N12
VCC1_5	U16
VCC1_5	M11
VCC1_5	T12
VCC1_5	H26
VCC1_5	V15
VCC1_5	R14
VCC1_5	N18
VCC1_5	N20
VCC1_5	N22
VCC1_5	P17
VCC1_5	P19
VCC1_5	P21
VCC1_5	P23
VCC1_5	R18
VCC1_5	R20
VCC1_5	R22
VCC1_5	T17
VCC1_5	T19
VCC1_5	T21
VCC1_5	T23
VCC1_5	U18
VCC1_5	U20
VCC1_5	U22
VCC1_5	V17
VCC1_5	V19
VCC1_5	V21
VCC1_5	V23
VCC1_5	Y13
VCC1_5	AB13
VCC1_5	W12
VCC1_5	AE13
VCC1_5	AD13
VCC1_5	AC12
VCC1_5	AA12
VCC1_5	AC14
VCC1_5	AC16
VCC1_5	AD15
VCC1_5	AA14
VCC1_5	AE14
VCC1_5	AA16
VCC1_5	W14
VCC1_5	W16
VCC1_5	Y15
VCC1_5	AB15
VCC1_5	AB23

Name	Ball #
VCC1_5	AB21
VCC1_5	AC18
VCC1_5	AD17
VCC1_5	AD19
VCC1_5	AA18
VCC1_5	AA20
VCC1_5	W18
VCC1_5	W20
VCC1_5	W22
VCC1_5	Y17
VCC1_5	Y19
VCC1_5	Y21
VCC1_5	AB17
VCC1_5	AB19
VCC1_5	AC20
VCC1_5	AC22
VCC1_5	AD21
VCC1_5	AD23
VCC1_5	AA22
VCC1_5	Y23
VCC1_5	AB27
VCC1_5	AD25
VCC1_5	AC27
VCC1_5	AB25
VCC1_5	AA25
VCC1_5	Y27
VCC1_5	Y25
VCC1_5	AD35
VCC1_5	AC31
VCC1_5	AB33
VCC1_5	AA36
VCC1_5	Y31
VCC1_5	AE31
VCC1_5	D1
VCC1_5	K9
VCC1_5	K8
VCC1_5	J8
VCC1_5	G4
VCC1_5	D16
VCC1_5	A16
VCC1_5	B18
VCC1_5	K16
VCC1_5	F15
VCC1_5	M16
VCC1_5	L18
VCC1_5	H18
VCC1_5	G19
VCC2_5	N11
VCC2_5	T27
VCC3_3	V28
VCC3_3	L33
VCC3_3	K31
VCC3_3	P26

Name	Ball #
VCC3_3	P25
VCC3_3	N26
VCC3_3	N24
VCC3_3	M29
VCC3_3	T26
VCC3_3	T24
VCC3_3	R29
VCC3_3	R27
VCC3_3	R25
VCC3_3	P27
VCC3_3	N31
VCC3_3	M35
VCC3_3	P36
VCC3_3	P33
VCC3_3	W24
VCC3_3	W36
VCC3_3	AH34
VCC3_3	C14
VCC3_3	L14
VCC3_3	L13
VCC3_3	K13
VCC3_3	H15
VCC3_3	E12
VCC3_3	M13
VCC3_3	A13
VCCSUS1_5	M27
VCCSUS1_5	N27
VCCSUS1_5	H10
VCCSUS3_3	C5
VCCSUS3_3	B3
VCCSUS3_3	A4
VCCSUS3_3	G7
VCCSUS3_3	E6
VCCSUS3_3	L26
VCCSUS3_3	L25
VCCSUS3_3	K28
VCCSUS3_3	K26
VCCSUS3_3	K25
VCCSUS3_3	H30
VCCSUS3_3	F32
VCCSUS3_3	M25
VCCSUS3_3	D34
VTT	P1
VTT	V6
VTT	U9
VTT	R5
VTT	P6
VTT	U12
VTT	R10
VTT	AC2
VTT	W1
VTT	AD5
VTT	AC9

Name	Ball #
VTT	W6
VTT	AB7
VTT	AB4
VTT	AE7
VTT	Y10
VTT	AE12
VTT	AE10
VTT	AD12
VTT	Y11
VTT	AB11
VTT	AG1
VTT	AL5
VTT	AJ3
VTT	AM7
VTT	AL3
VTT	AJ9
VTT	AJ6
VTT	AG8
VTT	AG5
VTT	AF3
VTT	AM10
VTT	AN1
VTT	AN9
VTT	AR6
VTT	AN4
VTT	AP9
VTT	T2
VCCRTC	C27
VCC_PEA_BG	AB28
VCC_PEA_PLL	AC26
VCC_PEB_BG	L11
VCC_PEB_PLL	M9
VCC_SATA_BG	J18
VCC_SATA_PLL	G26
VCC_USB_BG	K12
VCC_USB_PLL	F9
VSS	N25
VSS	M26
VSS	U26
VSS	U36
VSS	U33
VSS	T34
VSS	T31
VSS	R35
VSS	R32
VSS	N34
VSS	M32
VSS	W2
VSS	AC1
VSS	Y9
VSS	W5
VSS	AE9
VSS	AE6

Name	Ball #
VSS	AD4
VSS	AC8
VSS	AB6
VSS	AB3
VSS	Y12
VSS	W13
VSS	W11
VSS	AC13
VSS	AC11
VSS	AB12
VSS	AA13
VSS	AA11
VSS	AA15
VSS	AB14
VSS	AB16
VSS	AC15
VSS	AD14
VSS	AD16
VSS	AE15
VSS	W15
VSS	Y14
VSS	Y16
VSS	AA23
VSS	Y22
VSS	AA17
VSS	AA19
VSS	AA21
VSS	AB18
VSS	AB20
VSS	AB22
VSS	AC17
VSS	AC19
VSS	AC21
VSS	AD18
VSS	AD20
VSS	AD22
VSS	AE19
VSS	AE21
VSS	AE23
VSS	W17
VSS	W19
VSS	W21
VSS	W23
VSS	Y18
VSS	Y20
VSS	AC23
VSS	Y30
VSS	Y24
VSS	W26
VSS	W25
VSS	N28
VSS	AD30
VSS	AD27

Name	Ball #
VSS	AD24
VSS	AC28
VSS	AC25
VSS	AC24
VSS	AB29
VSS	AB26
VSS	AB24
VSS	AA28
VSS	AA24
VSS	Y34
VSS	W31
VSS	AE34
VSS	AD36
VSS	AD32
VSS	AD31
VSS	AC34
VSS	AB36
VSS	AB32
VSS	AA35
VSS	AA32
VSS	AA31
VSS	AJ1
VSS	AK1
VSS	AG2
VSS	AF1
VSS	AL2
VSS	AM3
VSS	AJ8
VSS	AJ5
VSS	AG7
VSS	AG4
VSS	AM9
VSS	AG14
VSS	AK13
VSS	AJ13
VSS	AJ11
VSS	AH15
VSS	AG13
VSS	AG10
VSS	AF16
VSS	AF14
VSS	AF12
VSS	AM13
VSS	AM11
VSS	AL10
VSS	AK19
VSS	AL15
VSS	AN24
VSS	AJ17
VSS	AG19
VSS	AF22
VSS	AF20
VSS	AF18

Name	Ball #
VSS	AM20
VSS	AL23
VSS	AK26
VSS	AJ28
VSS	AH24
VSS	AG26
VSS	AF30
VSS	AF24
VSS	AM28
VSS	AL32
VSS	AH35
VSS	AH31
VSS	AF36
VSS	AF33
VSS	AM35
VSS	AK33
VSS	AR2
VSS	AP2
VSS	AP1
VSS	AN8
VSS	AT6
VSS	AT4
VSS	AT3
VSS	AR9
VSS	AP4
VSS	AN7
VSS	AT13
VSS	AT10
VSS	AP15
VSS	AP11
VSS	AN13
VSS	AP18
VSS	AT19
VSS	AT17
VSS	AR19
VSS	AP22
VSS	AT24
VSS	AR27
VSS	AP30
VSS	AN25
VSS	AT34
VSS	AT31
VSS	AR35
VSS	AP36
VSS	AN36
VSS	AN32
VSS	C2
VSS	C1
VSS	B2
VSS	D7
VSS	D4
VSS	C8
VSS	A7

Name	Ball #
VSS	A3
VSS	B9
VSS	B6
VSS	D13
VSS	D10
VSS	C11
VSS	B15
VSS	B12
VSS	A10
VSS	D22
VSS	D19
VSS	A22
VSS	A19
VSS	C20
VSS	C17
VSS	B21
VSS	D28
VSS	D25
VSS	A28
VSS	A25
VSS	C29
VSS	C26
VSS	B30
VSS	B27
VSS	D31
VSS	A34
VSS	A31
VSS	C36
VSS	C35
VSS	C32
VSS	B35
VSS	B33
VSS	K1
VSS	J2
VSS	G1
VSS	F2
VSS	L9
VSS	L6
VSS	L3
VSS	K7
VSS	K4
VSS	J5
VSS	H9
VSS	H6
VSS	H3
VSS	F8
VSS	F5
VSS	E9
VSS	E3
VSS	L15
VSS	L12
VSS	K10
VSS	J14

Name	Ball #
VSS	J11
VSS	H12
VSS	G16
VSS	G13
VSS	G10
VSS	F14
VSS	F11
VSS	E15
VSS	L23
VSS	L21
VSS	K19
VSS	J23
VSS	J20
VSS	J17
VSS	H21
VSS	G22
VSS	F23
VSS	E21
VSS	E18
VSS	L30
VSS	L27
VSS	L24
VSS	J29
VSS	J26
VSS	H27
VSS	H24
VSS	G28
VSS	G25
VSS	F29
VSS	F26
VSS	E30
VSS	E27
VSS	E24
VSS	L36
VSS	K34
VSS	J35
VSS	J32
VSS	H36
VSS	H33
VSS	G34
VSS	G31
VSS	E36
VSS	E33
VSS	N2
VSS	T1
VSS	N1
VSS	V5
VSS	U8
VSS	R9
VSS	R4
VSS	P7
VSS	M8
VSS	M5

Name	Ball #
VSS	R13
VSS	V12
VSS	U13
VSS	R12
VSS	R11
VSS	N13
VSS	N10
VSS	U15
VSS	M14
VSS	N15
VSS	P16
VSS	R15
VSS	T14
VSS	T16
VSS	V14
VSS	V16
VSS	P14
VSS	M17
VSS	M19
VSS	M20
VSS	N17
VSS	N19
VSS	N21
VSS	N23
VSS	P18
VSS	P20
VSS	P22
VSS	R17
VSS	R19
VSS	R21
VSS	R23
VSS	T18
VSS	T20
VSS	T22
VSS	U17
VSS	U19
VSS	U21
VSS	U23
VSS	V18
VSS	V20
VSS	V22
VSS	M22
VSS	V25
VSS	V24
VSS	U30
VSS	U27
VSS	U24
VSS	T28
VSS	T25
VSS	R26
VSS	R24
VSS	P30
VSS_CORE_PLL	AG11

Name	Ball #
VSS_PEA_BG	Y26
VSS_PEA_PLL	AD26
VSS_PEB_BG	L10
VSS_SATA_BG	J19
VSS_USB_BG	K11
N/C	L22
N/C	AR3
N/C	AT9
N/C	AT8
N/C	AH19
N/C	D15
N/C	AH13
N/C	AJ4
N/C	AM4
N/C	AL4
N/C	AE17
N/C	AE16
N/C	AE8
N/C	W4
N/C	AB5
N/C	P24
N/C	R28
N/C	U25
N/C	U28
N/C	M10
N/C	M12
N/C	N8
N/C	N9
N/C	F35
N/C	M15
N/C	M21
N/C	M23
N/C	M24
N/C	F4
N/C	G6
N/C	H8
N/C	J10
N/C	F3
N/C	G5
N/C	H7
N/C	J9
N/C	D27
N/C	A8
N/C	B10
N/C	A9
N/C	B7
N/C	B8
N/C	A14
N/C	A11
N/C	B13
N/C	B14
N/C	A12
N/C	B11

Name	Ball #
N/C	AJ33
N/C	AJ31
N/C	E10
N/C	F10
N/C	E11
N/C	G11
N/C	F12
N/C	G12
N/C	D11
N/C	C12
N/C	C13
N/C	D12
N/C	E13
N/C	F13
N/C	E14
N/C	G14
N/C	G15
N/C	A15
N/C	G2
N/C	H4
N/C	J6
N/C	G8
N/C	G3
N/C	H5

Name	Ball #
N/C	J7
N/C	G9
N/C	T32
N/C	U34
N/C	R33
N/C	U31
N/C	U32
N/C	E35
N/C	L28
N/C	B31
N/C	P35
N/C	A5
N/C	A6
N/C	B4
N/C	B5
N/C	C3
N/C	C4
N/C	D2
N/C	D3
N/C	T10
N/C	AN10
N/C	AK11
N/C	AG27
N/C	AE25

able 129. Signal Listing — Numerically by Ball Number (Sheet 1 of 7)

Name	Ball#
VSS	A10
N/C	A11
N/C	A12
VCC3_3	A13
N/C	A14
N/C	A15
VCC1_5	A16
GPO[19]	A17
GPI[13]	A18
VSS	A19
LAD[0]/FWH[0]	A20
A20M#	A21
VSS	A22
IGNNE#	A23
FERR#	A24
VSS	A25
RSMRST#	A26
RTCX2	A27
VSS	A28
RTEST#	A29
VSS	A3
PIRQ[B]#	A30
VSS	A31
GPO[17]	A32
GPI[0]	A33

Name	Ball#
VSS	A34
VCCSUS3_3	A4
N/C	A5
N/C	A6
VSS	A7
N/C	A8
N/C	A9
HA[19]#	AA1
HA[15]#	AA10
VSS	AA11
VCC1_5	AA12
VSS	AA13
VCC1_5	AA14
VSS	AA15
VCC1_5	AA16
VSS	AA17
VCC1_5	AA18
VSS	AA19
HA[28]#	AA2
VCC1_5	AA20
VSS	AA21
VCC1_5	AA22
VSS	AA23
VSS	AA24
VCC1_5	AA25
PEA0_Rn[3]	AA26

Name	Ball#
PEA0_Rp[3]	AA27
VSS	AA28
PEA0_Tn[3]	AA29
HA[20]#	AA3
PEA0_Tp[3]	AA30
VSS	AA31
VSS	AA32
PEA0_Tn[0]	AA33
PEA0_Tp[0]	AA34
VSS	AA35
VCC1_5	AA36
HA[18]#	AA4
HA[24]#	AA5
HA[25]#	AA6
HREQ[0]#	AA7
HA[4]#	AA8
HA[8]#	AA9
HD[22]#	AB1
HREQ[3]#	AB10
VTT	AB11
VSS	AB12
VCC1_5	AB13
VSS	AB14
VCC1_5	AB15
VSS	AB16
VCC1_5	AB17

Name	Ball#
VSS	AB18
VCC1_5	AB19
HD[21]#	AB2
VSS	AB20
VCC1_5	AB21
VSS	AB22
VCC1_5	AB23
VSS	AB24
VCC1_5	AB25
VSS	AB26
VCC1_5	AB27
VCC_PEA_BG	AB28
VSS	AB29
VSS	AB3
PEA0_Rn[1]	AB30
PEA0_Rp[1]	AB31
VSS	AB32
VCC1_5	AB33
PEA0_Tn[1]	AB34
PEA0_Tp[1]	AB35
VSS	AB36
VTT	AB4
N/C	AB5
VSS	AB6
VTT	AB7
HA[9]#	AB8

Name	Ball#
HA[6]#	AB9
VSS	AC1
HREQ[1]#	AC10
VSS	AC11
VCC1_5	AC12
VSS	AC13
VCC1_5	AC14
VSS	AC15
VCC1_5	AC16
VSS	AC17
VCC1_5	AC18
VSS	AC19
VTT	AC2
VCC1_5	AC20
VSS	AC21
VCC1_5	AC22
VSS	AC23
VSS	AC24
VSS	AC25
VCC_PEA_PLL	AC26
VCC1_5	AC27
VSS	AC28
PEA0_Tn[7]	AC29
HD[16]#	AC3
PEA0_Tp[7]	AC30
VCC1_5	AC31
PEA_CLKn	AC32
PEA_CLKp	AC33
VSS	AC34
PEA_RCOMPO	AC35
PEA_ICOMPI	AC36
HD[17]#	AC4
HD[2]#	AC5
HD[7]#	AC6
HD[0]#	AC7
VSS	AC8
VTT	AC9
HD[23]#	AD1
HDVREF[0]	AD10
HD[13]#	AD11
VTT	AD12
VCC1_5	AD13
VSS	AD14
VCC1_5	AD15
VSS	AD16
VCC1_5	AD17
VSS	AD18
VCC1_5	AD19
HD[29]#	AD2
VSS	AD20
VCC1_5	AD21
VSS	AD22
VCC1_5	AD23

Name	Ball#
VSS	AD24
VCC1_5	AD25
VSS_PEA_PLL	AD26
VSS	AD27
PEA0_Rn[6]	AD28
PEA0_Rp[6]	AD29
HD[20]#	AD3
VSS	AD30
VSS	AD31
VSS	AD32
PEA0_Rn[4]	AD33
PEA0_Rp[4]	AD34
VCC1_5	AD35
VSS	AD36
VSS	AD4
VTT	AD5
HD[1]#	AD6
HD[4]#	AD7
HD[6]#	AD8
HA[3]#	AD9
HDSTBn[1]#	AE1
VTT	AE10
HD[9]#	AE11
VTT	AE12
VCC1_5	AE13
VCC1_5	AE14
VSS	AE15
N/C	AE16
N/C	AE17
VCC_DDR	AE18
VSS	AE19
HDSTBp[1]#	AE2
VCC_DDR	AE20
VSS	AE21
VCC_DDR	AE22
VSS	AE23
VCC_DDR	AE24
N/C	AE25
VCC_DDR	AE26
DEBUG[2]	AE27
TEST#	AE28
PEA0_Tn[6]	AE29
HDSTBp[0]#	AE3
PEA0_Tp[6]	AE30
VCC1_5	AE31
PEA0_Rn[7]	AE32
PEA0_Rp[7]	AE33
VSS	AE34
PEA0_Tn[4]	AE35
PEA0_Tp[4]	AE36
HDSTBn[0]#	AE4
HD[5]#	AE5
VSS	AE6

Name	Ball#
VTT	AE7
N/C	AE8
VSS	AE9
VSS	AF1
HD[3]#	AF10
VCC_DDR_PLL	AF11
VSS	AF12
VCC_DDR	AF13
VSS	AF14
VCC_DDR	AF15
VSS	AF16
VCC_DDR	AF17
VSS	AF18
VCC_DDR	AF19
DINV[1]#	AF2
VSS	AF20
VCC_DDR	AF21
VSS	AF22
VCC_DDR	AF23
VSS	AF24
VCC_DDR	AF25
CPU_SEL[1]	AF26
DEBUG[4]	AF27
TMS	AF28
DEBUG[6]	AF29
VTT	AF3
VSS	AF30
PEA0_Tn[5]	AF31
PEA0_Tp[5]	AF32
VSS	AF33
PEA0_Rn[5]	AF34
PEA0_Rp[5]	AF35
VSS	AF36
HD[18]#	AF4
HD[24]#	AF5
HD[31]#	AF6
HD[12]#	AF7
HD[25]#	AF8
HD[15]#	AF9
VTT	AG1
VSS	AG10
VSS_CORE_PLL	AG11
VCC_CORE_PLL	AG12
VSS	AG13
VSS	AG14
DDR_DQ[56]	AG15
DDR_CS[0]#	AG16
DDR_DQ[35]	AG17
DDR_DQSp[13]	AG18
VSS	AG19
VSS	AG2
VCC_DDR	AG20
DDR_MA[0]	AG21

Name	Ball#
DDR_DQ[19]	AG22
DDR_DQ[23]	AG23
DDR_VREF	AG24
DDR_MA[11]	AG25
VSS	AG26
N/C	AG27
PE_HPINTR#	AG28
PWRGD	AG29
DINV[0]#	AG3
RSTIN#	AG30
DEBUG[0]	AG31
TRST#	AG32
DEBUG[3]	AG33
DEBUG[7]	AG34
DEBUG[1]	AG35
TCK	AG36
VSS	AG4
VTT	AG5
HD[10]#	AG6
VSS	AG7
VTT	AG8
HD[11]#	AG9
HD[26]#	AH1
HCLKINn	AH10
HCLKINp	AH11
DDR_CS[7]#	AH12
N/C	AH13
DDR_DQ[59]	AH14
VSS	AH15
VCC_DDR	AH16
DDR_DQ[38]	AH17
DDR_DQSn[13]	AH18
N/C	AH19
HD[19]#	AH2
DDR_CB[3]	AH20
DDR_CMDCLKn[1]	AH21
DDR_MA[1]	AH22
DDR_DQ[18]	AH23
VSS	AH24
VCC_DDR	AH25
DDR_DQ[4]	AH26
DDR_DQ[5]	AH27
DDR_IMPCRES	AH28
DDR_RES2	AH29
HD[30]#	AH3
DEBUG[5]	AH30
VSS	AH31
VCC_DDR	AH32
TDO	AH33
VCC3_3	AH34
VSS	AH35
VCC_DDR	AH36
HD[27]#	AH4

Name	Ball#
HD[28]#	AH5
HD[8]#	AH6
HD[14]#	AH7
HD[47]#	AH8
HD[33]#	AH9
VSS	AJ1
HDVREF[1]	AJ10
VSS	AJ11
VCC_DDR	AJ12
VSS	AJ13
DDR_DQ[63]	AJ14
DDR_CS[6]#	AJ15
DDR_CS[2]#	AJ16
VSS	AJ17
VCC_DDR	AJ18
DDR_DQ[41]	AJ19
HD[38]#	AJ2
DDR_BA[1]	AJ20
DDR_CMDCLKp[1]	AJ21
VCC_DDR	AJ22
DDR_MA[3]	AJ23
VCC_DDR	AJ24
DDR_MA[6]	AJ25
DDR_CKE[2]	AJ26
DDR_DQ[0]	AJ27
VSS	AJ28
VCC_DDR	AJ29
VTT	AJ3
DDR_BA[2]	AJ30
N/C	AJ31
SMBSDA	AJ32
N/C	AJ33
TDI	AJ34
SMBSCL	AJ35
DDR_CRES0	AJ36
N/C	AJ4
VSS	AJ5
VTT	AJ6
HD[48]#	AJ7
VSS	AJ8
VTT	AJ9
VSS	AK1
HD[46]#	AK10
N/C	AK11
DDR_DQ[50]	AK12
VSS	AK13
VCC_DDR	AK14
DDR_CS[1]#	AK15
DDR_DQ[43]	AK16
DDR_DQ[42]	AK17
DDR_DQSn[4]	AK18
VSS	AK19
HD[34]#	AK2

Name	Ball#
VCC_DDR	AK20
DDR_CB[2]	AK21
DDR_CB[6]	AK22
DDR_MA[2]	AK23
DDR_DQ[22]	AK24
DDR_DQ[17]	AK25
VSS	AK26
VCC_DDR	AK27
DDR_DQ[1]	AK28
DDR_DQSp[9]	AK29
HD[41]#	AK3
DDR_DQSn[9]	AK30
DDR_SLWCRES	AK31
DDR_CKE[0]	AK32
VSS	AK33
VCC_DDR	AK34
DDR_CKE[3]	AK35
DDR_RES1	AK36
HD[42]#	AK4
DINV[2]#	AK5
HDSTBn[2]#	AK6
HD[43]#	AK7
HD[58]#	AK8
HD[40]#	AK9
HD[39]#	AL1
VSS	AL10
DDR_CS[5]#	AL11
DDR_DQ[55]	AL12
DDR_DQ[62]	AL13
DDR_DQ[57]	AL14
VSS	AL15
VCC_DDR	AL16
DDR_DQ[39]	AL17
DDR_DQSp[4]	AL18
DDR_DQ[40]	AL19
VSS	AL2
DDR_BA[0]	AL20
DDR_CB[7]	AL21
DDR_DQSp[8]	AL22
VSS	AL23
VCC_DDR	AL24
DDR_DQSp[2]	AL25
DDR_DQ[21]	AL26
DDR_MA[5]	AL27
DDR_CKE[1]	AL28
DDR_DQSn[0]	AL29
VTT	AL3
DDR_DQSp[0]	AL30
VCC_DDR	AL31
VSS	AL32
DDR_MA[14]	AL33
DDR_MA[12]	AL34
CPU_SEL[0]	AL35

Name	Ball#
CPU_SEL[2]	AL36
N/C	AL4
VTT	AL5
HDSTBp[2]#	AL6
HD[57]#	AL7
HD[62]#	AL8
HD[32]#	AL9
HD[37]#	AM1
VTT	AM10
VSS	AM11
VCC_DDR	AM12
VSS	AM13
DDR_DQ[49]	AM14
DDR_CS[3]#	AM15
DDR_CS[4]#	AM16
DDR_CAS#	AM17
DDR_DQSp[5]	AM18
DDR_DQ[45]	AM19
HD[35]#	AM2
VSS	AM20
VCC_DDR	AM21
DDR_DQSn[8]	AM22
DDR_CB[1]	AM23
DDR_CB[4]	AM24
DDR_DQSn[2]	AM25
DDR_DQ[16]	AM26
DDR_DQ[20]	AM27
VSS	AM28
VCC_DDR	AM29
VSS	AM3
DDR_DQ[6]	AM30
DDR_DQ[7]	AM31
DDR_DQ[2]	AM32
DDR_DQ[12]	AM33
DDR_DQ[13]	AM34
VSS	AM35
VCC_DDR	AM36
N/C	AM4
HD[52]#	AM5
HD[50]#	AM6
VTT	AM7
HD[55]#	AM8
VSS	AM9
VTT	AN1
N/C	AN10
DDR_DQ[51]	AN11
DDR_DQSn[6]	AN12
VSS	AN13
VCC_DDR	AN14
DDR_DQ[61]	AN15
DDR_DQ[60]	AN16
DDR_DQ[34]	AN17
DDR_DQSn[5]	AN18

Name	Ball#
DDR_DQ[44]	AN19
HD[45]#	AN2
VCC_DDR	AN20
DDR_DQ[36]	AN21
DDR_DQSn[17]	AN22
DDR_DQSp[17]	AN23
VSS	AN24
VSS	AN25
DDR_DQSn[11]	AN26
DDR_DQ[30]	AN27
DDR_MA[8]	AN28
DDR_DQ[8]	AN29
HD[51]#	AN3
DDR_DQ[9]	AN30
DDR_DQ[3]	AN31
VSS	AN32
VCC_DDR	AN33
DDR_DQSn[10]	AN34
DDR_DQSp[10]	AN35
VSS	AN36
VTT	AN4
HD[44]#	AN5
HDSTBp[3]#	AN6
VSS	AN7
VSS	AN8
VTT	AN9
VSS	AP1
DDR_MA[13]	AP10
VSS	AP11
VCC_DDR	AP12
DDR_DQSn[16]	AP13
DDR_DQSn[15]	AP14
VSS	AP15
VCC_DDR	AP16
DDR_DQ[47]	AP17
VSS	AP18
DDR_WE#	AP19
VSS	AP2
DDR_RAS#	AP20
DDR_MA[10]	AP21
VSS	AP22
VCC_DDR	AP23
DDR_CB[5]	AP24
DDR_CMDCLKn[2]	AP25
DDR_DQSp[11]	AP26
DDR_DQSp[3]	AP27
DDR_DQ[25]	AP28
DDR_DQ[29]	AP29
DINV[3]#	AP3
VSS	AP30
VCC_DDR	AP31
DDR_MA[7]	AP32
DDR_DQ[14]	AP33

Name	Ball#
DDR_DQSp[1]	AP34
DDR_DQSn[1]	AP35
VSS	AP36
VSS	AP4
HD[49]#	AP5
HDSTBn[3]#	AP6
HD[60]#	AP7
HD[61]#	AP8
VTT	AP9
DDR_DQ[58]	AR10
DDR_DQSn[7]	AR11
DDR_DQSp[6]	AR12
DDR_DQSp[16]	AR13
DDR_DQSp[15]	AR14
DDR_DQ[53]	AR15
DDR_DQ[52]	AR16
DDR_DQ[46]	AR17
DDR_DQSp[14]	AR18
VSS	AR19
VSS	AR2
DDR_DQ[33]	AR20
DDR_DQ[32]	AR21
DDR_CMDCLKn[3]	AR22
DDR_CMDCLKp[3]	AR23
DDR_CB[0]	AR24
DDR_CMDCLKp[2]	AR25
DDR_DQ[26]	AR26
VSS	AR27
DDR_DQSn[3]	AR28
DDR_DQSn[12]	AR29
N/C	AR3
DDR_MA[4]	AR30
DDR_DQ[28]	AR31
DDR_DQ[10]	AR32
DDR_DQ[15]	AR33
DDR_MA[9]	AR34
VSS	AR35
HD[36]#	AR4
HD[53]#	AR5
VTT	AR6
HD[63]#	AR7
HD[56]#	AR8
VSS	AR9
VSS	AT10
DDR_DQSp[7]	AT11
DDR_DQ[54]	AT12
VSS	AT13
VCC_DDR	AT14
DDR_DQ[48]	AT15
VCC_DDR	AT16
VSS	AT17

Name	Ball#
DDR_DQSn[14]	AT18
VSS	AT19
VCC_DDR	AT20
DDR_DQ[37]	AT21
DDR_CMDCLKp[0]	AT22
DDR_CMDCLKn[0]	AT23
VSS	AT24
VCC_DDR	AT25
DDR_DQ[27]	AT26
DDR_DQ[31]	AT27
VCC_DDR	AT28
DDR_DQSp[12]	AT29
VSS	AT3
DDR_DQ[24]	AT30
VSS	AT31
VCC_DDR	AT32
DDR_DQ[11]	AT33
VSS	AT34
VSS	AT4
HD[54]#	AT5
VSS	AT6
HD[59]#	AT7
N/C	AT8
N/C	AT9
N/C	B10
N/C	B11
VSS	B12
N/C	B13
N/C	B14
VSS	B15
GPO[20]	B16
GPO[18]	B17
VCC1_5	B18
LAD[2]/FWH[2]	B19
VSS	B2
LAD[1]/FWH[1]	B20
VSS	B21
NMI	B22
SMI#	B23
VCC_CPU	B24
PWROK	B25
RTCX1	B26
VSS	B27
GPIO[27]	B28
PIRQ[C]#	B29
VCCSUS3_3	B3
VSS	B30
N/C	B31
THRMTRIP#	B32
VSS	B33
CLK14	B34
VSS	B35

Name	Ball#
N/C	B4
N/C	B5
VSS	B6
N/C	B7
N/C	B8
VSS	B9
VSS	C1
USBp[0]	C10
VSS	C11
N/C	C12
N/C	C13
VCC3_3	C14
GPO[23]	C15
GPO[21]	C16
VSS	C17
GPI[30]	C18
LAD[3]/FWH[3]	C19
VSS	C2
VSS	C20
INTR	C21
STPCLK#	C22
VCC_CPU	C23
RCIN#	C24
INTVRMEN	C25
VSS	C26
VCCRTC	C27
INTRUDER#	C28
VSS	C29
N/C	C3
PIRQ[D]#	C30
PIRQ[A]#	C31
VSS	C32
SPKR	C33
PCIRST#	C34
VSS	C35
VSS	C36
N/C	C4
VCCSUS3_3	C5
USBn[3]	C6
USBp[3]	C7
VSS	C8
USBn[0]	C9
VCC1_5	D1
VSS	D10
N/C	D11
N/C	D12
VSS	D13
V5REF	D14
N/C	D15
VCC1_5	D16
GPIO[34]	D17
GPI[26]	D18
VSS	D19

Name	Ball#
N/C	D2
LFRAME#/FWH[4]	D20
CPUSLP#	D21
VSS	D22
INIT#	D23
VCC_CPU	D24
VSS	D25
GPIO[24]	D26
N/C	D27
VSS	D28
SLP_S3#	D29
N/C	D3
SMLINK[0]	D30
VSS	D31
PIRQ[E]#/GPI[2]	D32
GNT[0]#	D33
VCCSUS3_3	D34
GPI[1]	D35
GPO[16]	D36
VSS	D4
OC[0]#	D5
OC[1]#	D6
VSS	D7
USBn[1]	D8
USBp[1]	D9
OC[3]#	E1
N/C	E10
N/C	E11
VCC3_3	E12
N/C	E13
N/C	E14
VSS	E15
GPIO[33]	E16
GPI[7]	E17
VSS	E18
GPI[31]	E19
OC[2]#	E2
LDRQ[0]#	E20
VSS	E21
A20GATE	E22
INIT3_3V#	E23
VSS	E24
WAKE#	E25
SMBALERT#\GPI[11]	E26
VSS	E27
WL_PU[1]	E28
SLP_S4#	E29
VSS	E3
VSS	E30
PIRQ[F]#/GPI[3]	E31

Name	Ball#
PIRQ[H]#/ GPI[5]	E32
VSS	E33
REQ[1]#	E34
N/C	E35
VSS	E36
GPI[9]	E4
V5REFSUS	E5
VCCSUS3_3	E6
USBn[2]	E7
USBp[2]	E8
VSS	E9
GPI[10]	F1
N/C	F10
VSS	F11
N/C	F12
N/C	F13
VSS	F14
VCC1_5	F15
GPIO[32]	F16
VCC1_5	F17
GPI[29]	F18
WL_PU[0]	F19
VSS	F2
VCC1_5	F20
LDRQ[1]#/ GPI[41]	F21
CPUPWRGD/ GPO[49]	F22
VSS	F23
THRM#	F24
SUSCLK	F25
VSS	F26
GPIO[25]	F27
PWRBTN#	F28
VSS	F29
N/C	F3
SMLINK[1]	F30
PIRQ[G]#/ GPI[4]	F31
VCCSUS3_3	F32
AD[30]	F33
AD[29]	F34
N/C	F35
DEVSEL#	F36
N/C	F4
VSS	F5
GPI[14]	F6
GPI[15]	F7
VSS	F8
VCC_USB_PLL	F9
VSS	G1
VSS	G10
N/C	G11

Name	Ball#
N/C	G12
VSS	G13
N/C	G14
N/C	G15
VSS	G16
GPI[6]	G17
GPI[12]	G18
VCC1_5	G19
N/C	G2
SATA_RXn[3]	G20
SATA_RXp[3]	G21
VSS	G22
SATA_CLKn	G23
SATA_CLKp	G24
VSS	G25
VCC_SATA_PLL	G26
GPI[8]	G27
VSS	G28
SLP_S5#	G29
N/C	G3
SMBCLK	G30
VSS	G31
AD[31]	G32
AD[27]	G33
VSS	G34
STOP#	G35
PERR#	G36
VCC1_5	G4
N/C	G5
N/C	G6
VCCSUS3_3	G7
N/C	G8
N/C	G9
PEB0_Tn[1]	H1
VCCSUS1_5	H10
CLK48	H11
VSS	H12
SATA_TXp[0]	H13
SATA_TXn[0]	H14
VCC3_3	H15
SATA_RXn[1]	H16
SATA_RXp[1]	H17
VCC1_5	H18
SATA_TXp[3]	H19
PEB0_Tp[1]	H2
SATA_TXn[3]	H20
VSS	H21
SATA_RXn[5]	H22
SATA_RXp[5]	H23
VSS	H24
SATA_RBIASn	H25
VCC1_5	H26
VSS	H27

Name	Ball#
SUS_STAT#/ LPCPD#	H28
SMBDATA	H29
VSS	H3
VCCSUS3_3	H30
REQ[0]#	H31
AD[28]	H32
VSS	H33
AD[24]	H34
PLOCK#	H35
VSS	H36
N/C	H4
N/C	H5
VSS	H6
N/C	H7
N/C	H8
VSS	H9
PEB_RCOMPO	J1
N/C	J10
VSS	J11
USB_RBIASp	J12
USB_RBIASn	J13
VSS	J14
SATA_TXp[1]	J15
SATA_TXn[1]	J16
VSS	J17
VCC_SATA_BG	J18
VSS_SATA_BG	J19
VSS	J2
VSS	J20
SATA_TXp[4]	J21
SATA_TXn[4]	J22
VSS	J23
SERIRQ	J24
SATA_RBIASp	J25
VSS	J26
RI#	J27
GPIO[28]	J28
VSS	J29
PEB0_Rn[1]	J3
AD[25]	J30
AD[26]	J31
VSS	J32
C/BE[3]#	J33
AD[14]	J34
VSS	J35
SERR#	J36
PEB0_Rp[1]	J4
VSS	J5
N/C	J6
N/C	J7
VCC1_5	J8
N/C	J9
VSS	K1

Name	Ball#
VSS	K10
VSS_USB_BG	K11
VCC_USB_BG	K12
VCC3_3	K13
SATA_RXn[0]	K14
SATA_RXp[0]	K15
VCC1_5	K16
SATA_RXn[2]	K17
SATA_RXp[2]	K18
VSS	K19
PEB0_Tn[2]	K2
SATA_RXn[4]	K20
SATA_RXp[4]	K21
VRMPWRGD/ VGATE	K22
SATA_LED#	K23
SYS_RESET#	K24
VCCSUS3_3	K25
VCCSUS3_3	K26
VCC1_5	K27
VCCSUS3_3	K28
C/BE[0]#	K29
PEB0_Tp[2]	K3
GPI[40]	K30
VCC3_3	K31
AD[13]	K32
AD[12]	K33
VSS	K34
PAR	K35
AD[15]	K36
VSS	K4
PEB0_Rp[2]	K5
PEB0_Rn[2]	K6
VSS	K7
VCC1_5	K8
VCC1_5	K9
PEB_CLKn	L1
VSS_PEB_BG	L10
VCC_PEB_BG	L11
VSS	L12
VCC3_3	L13
VCC3_3	L14
VSS	L15
SATA_TXp[2]	L16
SATA_TXn[2]	L17
VCC1_5	L18
SATA_TXp[5]	L19
PEB_ICOMPI	L2
SATA_TXn[5]	L20
VSS	L21
N/C	L22
VSS	L23
VSS	L24
VCCSUS3_3	L25

Name	Ball#
VCCSUS3_3	L26
VSS	L27
N/C	L28
AD[7]	L29
VSS	L3
VSS	L30
AD[22]	L31
AD[23]	L32
VCC3_3	L33
AD[10]	L34
C/BE[1]#	L35
VSS	L36
PEB0_Tp[3]	L4
PEB0_Tn[3]	L5
VSS	L6
PEB0_Rp[0]	L7
PEB0_Rn[0]	L8
VSS	L9
PEB_CLKp	M1
N/C	M10
VCC1_5	M11
N/C	M12
VCC3_3	M13
VSS	M14
N/C	M15
VCC1_5	M16
VSS	M17
VCC1_5	M18
VSS	M19
VCC1_5	M2
VSS	M20
N/C	M21
VSS	M22
N/C	M23
N/C	M24
VCCSUS3_3	M25
VSS	M26
VCCSUS1_5	M27
PCICLK	M28
VCC3_3	M29
PEB0_Rp[3]	M3
AD[20]	M30
AD[19]	M31
VSS	M32
AD[5]	M33
AD[6]	M34
VCC3_3	M35
AD[11]	M36
PEB0_Rn[3]	M4
VSS	M5
PEB0_Tp[0]	M6
PEB0_Tn[0]	M7
VSS	M8

Name	Ball#
VCC_PEB_PLL	M9
VSS	N1
VSS	N10
VCC2_5	N11
VCC1_5	N12
VSS	N13
VCC1_5	N14
VSS	N15
VCC1_5	N16
VSS	N17
VCC1_5	N18
VSS	N19
VSS	N2
VCC1_5	N20
VSS	N21
VCC1_5	N22
VSS	N23
VCC3_3	N24
VSS	N25
VCC3_3	N26
VCCSUS1_5	N27
VSS	N28
IRDY#	N29
ADS#	N3
C/BE[2]#	N30
VCC3_3	N31
AD[4]	N32
AD[3]	N33
VSS	N34
AD[9]	N35
AD[8]	N36
VCC1_5	N4
RSP#	N5
HCRES0	N6
VCC1_5	N7
N/C	N8
N/C	N9
VTT	P1
VCC1_5	P10
VCC1_5	P11
VCC1_5	P12
VCC1_5	P13
VSS	P14
VCC1_5	P15
VSS	P16
VCC1_5	P17
VSS	P18
VCC1_5	P19
DBSY#	P2
VSS	P20
VCC1_5	P21
VSS	P22
VCC1_5	P23

Name	Ball#
N/C	P24
VCC3_3	P25
VCC3_3	P26
VCC3_3	P27
TRDY#	P28
GNT[1]#	P29
DRDY#	P3
VSS	P30
PME#	P31
PLTRST#	P32
VCC3_3	P33
GPO[48]	P34
N/C	P35
VCC3_3	P36
BREQ[0]#	P4
CPURST#	P5
VTT	P6
VSS	P7
HTRDY#	P8
HSLWCRES	P9
BREQ[1]#	R1
VTT	R10
VSS	R11
VSS	R12
VSS	R13
VCC1_5	R14
VSS	R15
VCC1_5	R16
VSS	R17
VCC1_5	R18
VSS	R19
DP[0]#	R2
VCC1_5	R20
VSS	R21
VCC1_5	R22
VSS	R23
VSS	R24
VCC3_3	R25
VSS	R26
VCC3_3	R27
N/C	R28
VCC3_3	R29
BNR#	R3
AD[17]	R30
PEB_RPC[1]	R31
VSS	R32
N/C	R33
AD[21]	R34
VSS	R35
AD[2]	R36
VSS	R4
VTT	R5
DP[1]#	R6

Name	Ball#
BINIT#	R7
AP[1]#	R8
VSS	R9
VSS	T1
N/C	T10
HODTCRES	T11
VCC1_5	T12
VCC1_5	T13
VSS	T14
VCC1_5	T15
VSS	T16
VCC1_5	T17
VSS	T18
VCC1_5	T19
VTT	T2
VSS	T20
VCC1_5	T21
VSS	T22
VCC1_5	T23
VCC3_3	T24
VSS	T25
VCC3_3	T26
VCC2_5	T27
VSS	T28
FRAME#	T29
RS[0]#	T3
AD[16]	T30
VSS	T31
N/C	T32
PEB_RPC[0]	T33
VSS	T34
AD[1]	T35
AD[0]	T36
RS[1]#	T4
MCERR#	T5
RS[2]#	T6
HITM#	T7
HLOCK#	T8
DEFER#	T9
HA[34]#	U1
BPRI#	U10
HIT#	U11
VTT	U12
VSS	U13
VCC1_5	U14
VSS	U15
VCC1_5	U16
VSS	U17
VCC1_5	U18
VSS	U19
HA[35]#	U2
VCC1_5	U20
VSS	U21

Name	Ball#
VCC1_5	U22
VSS	U23
VSS	U24
N/C	U25
VSS	U26
VSS	U27
N/C	U28
V5REF	U29
HA[32]#	U3
VSS	U30
N/C	U31
N/C	U32
VSS	U33
N/C	U34
AD[18]	U35
VSS	U36
DP[2]#	U4
HADSTB[0]#	U5
HA[7]#	U6
HA[11]#	U7
VSS	U8
VTT	U9
HA[33]#	V1
AP[0]#	V10
DP[3]#	V11
VSS	V12
VCC1_5	V13
VSS	V14
VCC1_5	V15
VSS	V16
VCC1_5	V17
VSS	V18
VCC1_5	V19
HA[17]#	V2
VSS	V20
VCC1_5	V21
VSS	V22
VCC1_5	V23
VSS	V24
VSS	V25
SIU_DTR[1]#	V26
SIU_RXD[1]	V27
VCC3_3	V28
SIU_RI[1]#	V29
HA[31]#	V3
SIU_RXD[2]	V30
SIU_RTS[1]#	V31
SIU_DCD[2]#	V32
SIU_TXD[2]	V33
SIU_DCD[1]#	V34
SIU_TXD[1]	V35
SIU_CTS[1]#	V36
HA[29]#	V4

Name	Ball#
VSS	V5
VTT	V6
HA[10]#	V7
HA[12]#	V8
HACVREF	V9
VTT	W1
HA[16]#	W10
VSS	W11
VCC1_5	W12
VSS	W13
VCC1_5	W14
VSS	W15
VCC1_5	W16
VSS	W17
VCC1_5	W18
VSS	W19
VSS	W2
VCC1_5	W20
VSS	W21
VCC1_5	W22
VSS	W23
VCC3_3	W24
VSS	W25
VSS	W26
WDT_TOUT#	W27
SIU_RI[2]#	W28
SIU_DTR[2]#	W29
HA[27]#	W3
UART_CLK	W30
VSS	W31
SIU_RTS[2]#	W32
SIU_DSR[2]#	W33
SIU_CTS[2]#	W34
SIU_DSR[1]#	W35
VCC3_3	W36
N/C	W4
VSS	W5
VTT	W6
HREQ[4]#	W7
HA[5]#	W8
HA[14]#	W9
HA[21]#	Y1
VTT	Y10
VTT	Y11
VSS	Y12
VCC1_5	Y13
VSS	Y14
VCC1_5	Y15
VSS	Y16
VCC1_5	Y17
VSS	Y18
VCC1_5	Y19
HA[23]#	Y2

Name	Ball#
VSS	Y20
VCC1_5	Y21
VSS	Y22
VCC1_5	Y23
VSS	Y24
VCC1_5	Y25
VSS_PEA_BG	Y26
VCC1_5	Y27
PEA0_Rn[0]	Y28
PEA0_Rp[0]	Y29
HADSTB[1]#	Y3
VSS	Y30
VCC1_5	Y31
PEA0_Rn[2]	Y32
PEA0_Rp[2]	Y33
VSS	Y34
PEA0_Tn[2]	Y35
PEA0_Tp[2]	Y36
HA[22]#	Y4
HA[30]#	Y5
HA[26]#	Y6
HREQ[2]#	Y7
HA[13]#	Y8
VSS	Y9

11.2 Interface Trace Length Compensation

This section details information about the internal component package trace lengths (see Table 130) to enable trace length compensation. Trace length compensation is required for platform design. The Platform Design Guides consider these package lengths in providing platform design guidance. Note that these lengths represent the actual lengths from pad to ball.

Table 130. System Bus Package Trace Length (Sheet 1 of 5)

Signal	Ball #	L_{PKG} (mils)
DINV[3]#	AP3	787.685
DINV[2]#	AK5	573.874
DINV[1]#	AF2	662.146
DINV[0]#	AG3	729.227
HA[35]#	U2	627.952
HA[34]#	U1	657.239
HA[33]#	V1	644.656
HA[32]#	U3	555.36
HA[31]#	V3	713.884
HA[30]#	Y5	495.001
HA[29]#	V4	523.324
HA[28]#	AA2	600.405
HA[27]#	W3	536.397
HA[26]#	Y6	411.327
HA[25]#	AA6	396.973
HA[24]#	AA5	459.053
HA[23]#	Y2	589.516
HA[22]#	Y4	505.643
HA[21]#	Y1	646.131
HA[20]#	AA3	549.98
HA[19]#	AA1	625.959
HA[18]#	AA4	501.989
HA[17]#	V2	615.715
HA[16]#	W10	372.297
HA[15]#	AA10	285.168
HA[14]#	W9	413.134
HA[13]#	Y8	349.839
HA[12]#	V8	497.464
HA[11]#	U7	522.713
HA[10]#	V7	485.616
HA[9]#	AB8	377.317
HA[8]#	AA9	316.326
HA[7]#	U6	647.853
HA[6]#	AB9	399.037
HA[5]#	W8	330.987
HA[4]#	AA8	444.73
HA[3]#	AD9	402.934
HADSTB[1]#	Y3	706.042
HADSTB[0]#	U5	712.96
HCLKINn	AH10	400.784
HCLKINp	AH11	401.208
HD[63]#	AR7	641.466
HD[62]#	AL8	553.247
HD[61]#	AP8	773.555
HD[60]#	AP7	596.506
HD[59]#	AT7	664.781
HD[58]#	AK8	396.061
HD[57]#	AL7	464.519
HD[56]#	AR8	733.407
HD[55]#	AM8	702.774
HD[54]#	AT5	717.09
HD[53]#	AR5	769.136
HD[52]#	AM5	657.01
HD[51]#	AN3	758.369
HD[50]#	AM6	750.091
HD[49]#	AP5	819.643
HD[48]#	AJ7	488.748
HD[47]#	AH8	394.842
HD[46]#	AK10	429.291
HD[45]#	AN2	923.458
HD[44]#	AN5	685
HD[43]#	AK7	504.718
HD[42]#	AK4	701.593
HD[41]#	AK3	727.402
HD[40]#	AK9	545.441
HD[39]#	AL1	798.166
HD[38]#	AJ2	708.395
HD[37]#	AM1	768.96
HD[36]#	AR4	897.863
HD[35]#	AM2	858.006
HD[34]#	AK2	782.976
HD[33]#	AH9	439.776
HD[32]#	AL9	540.113
HD[31]#	AF6	479.846
HD[30]#	AH3	745.126
HD[29]#	AD2	654.162
HD[28]#	AH5	668.479
HD[27]#	AH4	641.999
HD[26]#	AH1	827.104
HD[25]#	AF8	434.924
HD[24]#	AF5	506.564
HD[23]#	AD1	664.027
HD[22]#	AB1	658.874
HD[21]#	AB2	617.257

Signal	Ball #	L_{PKG} (mils)
HD[20]#	AD3	581.015
HD[19]#	AH2	884.657
HD[18]#	AF4	581.587
HD[17]#	AC4	533.158
HD[16]#	AC3	585.046
HD[15]#	AF9	456.377
HD[14]#	AH7	477.718
HD[13]#	AD11	423.797
HD[12]#	AF7	387.252
HD[11]#	AG9	453.2
HD[10]#	AG6	747.844
HD[9]#	AE11	335.195
HD[8]#	AH6	553.059
HD[7]#	AC6	392.576
HD[6]#	AD8	333.624
HD[5]#	AE5	461.619
HD[4]#	AD7	372.91
HD[3]#	AF10	473.521
HD[2]#	AC5	449.208
HD[1]#	AD6	511.718
HD[0]#	AC7	386.386
HDSTBn[3]#	AP6	669.959
HDSTBn[2]#	AK6	590.904
HDSTBn[1]#	AE1	688.542
HDSTBn[0]#	AE4	550.529
HDSTBp[3]#	AN6	671.808
HDSTBp[2]#	AL6	589.407
HDSTBp[1]#	AE2	687.38
HDSTBp[0]#	AE3	550.722
HREQ[4]#	W7	541.717
HREQ[3]#	AB10	427.394
HREQ[2]#	Y7	422.039
HREQ[1]#	AC10	270.23
HREQ[0]#	AA7	496.773
DDR_MA[14]	AL33	598.285
DDR_MA[13]	AP10	661.226
DDR_MA[12]	AL34	641.246
DDR_MA[11]	AG25	298.804
DDR_MA[10]	AP21	541.235
DDR_MA[9]	AR34	892.219
DDR_MA[8]	AN28	624.87
DDR_MA[7]	AP32	754.061
DDR_MA[6]	AJ25	338.381
DDR_MA[5]	AL27	557.82
DDR_MA[4]	AR30	748.478

Signal	Ball #	L_{PKG} (mils)
DDR_MA[3]	AJ23	358.391
DDR_MA[2]	AK23	414.525
DDR_MA[1]	AH22	351.613
DDR_MA[0]	AG21	197.665
DDR_BA[2]	AJ30	479.38
DDR_BA[1]	AJ20	445.537
DDR_BA[0]	AL20	471.378
DDR_DQ[63]	AJ14	671.92
DDR_DQ[62]	AL13	670.302
DDR_DQ[61]	AN15	671.385
DDR_DQ[60]	AN16	670.535
DDR_DQ[59]	AH14	665.298
DDR_DQ[58]	AR10	667.514
DDR_DQ[57]	AL14	667.063
DDR_DQ[56]	AG15	669.488
DDR_DQ[55]	AL12	755.711
DDR_DQ[54]	AT12	756.704
DDR_DQ[53]	AR15	752.216
DDR_DQ[52]	AR16	752.277
DDR_DQ[51]	AN11	754.049
DDR_DQ[50]	AK12	749.072
DDR_DQ[49]	AM14	752.955
DDR_DQ[48]	AT15	751.037
DDR_DQ[47]	AP17	653.299
DDR_DQ[46]	AR17	654.15
DDR_DQ[45]	AM19	656.211
DDR_DQ[44]	AN19	654.703
DDR_DQ[43]	AK16	654.947
DDR_DQ[42]	AK17	657.166
DDR_DQ[41]	AJ19	656.156
DDR_DQ[40]	AL19	655.274
DDR_DQ[39]	AL17	876.95
DDR_DQ[38]	AH17	884.367
DDR_DQ[37]	AT21	883.116
DDR_DQ[36]	AN21	884.125
DDR_DQ[35]	AG17	882.179
DDR_DQ[34]	AN17	881.849
DDR_DQ[33]	AR20	885.282
DDR_DQ[32]	AR21	886.131
DDR_DQ[31]	AT27	802.417
DDR_DQ[30]	AN27	801.672
DDR_DQ[29]	AP29	801.516
DDR_DQ[28]	AR31	803.302
DDR_DQ[27]	AT26	800.33
DDR_DQ[26]	AR26	800.286

Signal	Ball #	L_{PKG} (mils)
DDR_DQ[25]	AP28	804.784
DDR_DQ[24]	AT30	805.883
DDR_DQ[23]	AG23	567.645
DDR_DQ[22]	AK24	567.694
DDR_DQ[21]	AL26	567.589
DDR_DQ[20]	AM27	567.542
DDR_DQ[19]	AG22	568.959
DDR_DQ[18]	AH23	568.467
DDR_DQ[17]	AK25	568.265
DDR_DQ[16]	AM26	570.474
DDR_DQ[15]	AR33	947.168
DDR_DQ[14]	AP33	952.064
DDR_DQ[13]	AM34	946.36
DDR_DQ[12]	AM33	947.8
DDR_DQ[11]	AT33	952.959
DDR_DQ[10]	AR32	951.271
DDR_DQ[9]	AN30	951.689
DDR_DQ[8]	AN29	951.46
DDR_DQ[7]	AM31	741.031
DDR_DQ[6]	AM30	741.178
DDR_DQ[5]	AH27	742.174
DDR_DQ[4]	AH26	740.418
DDR_DQ[3]	AN31	740.528
DDR_DQ[2]	AM32	742.476
DDR_DQ[1]	AK28	742.402
DDR_DQ[0]	AJ27	740.465
DDR_CB[7]	AL21	646.94
DDR_CB[6]	AK22	645.865
DDR_CB[5]	AP24	645.663
DDR_CB[4]	AM24	648.752
DDR_CB[3]	AH20	649.747
DDR_CB[2]	AK21	649.521
DDR_CB[1]	AM23	648.26
DDR_CB[0]	AR24	648.074
DDR_RAS#	AP20	512.191
DDR_CAS#	AM17	556.939
DDR_WE#	AP19	600.75
DDR_CS[7]#	AH12	272.489
DDR_CS[6]#	AJ15	279.913
DDR_CS[5]#	AL11	452.11
DDR_CS[4]#	AM16	480.969
DDR_CS[3]#	AM15	406.953
DDR_CS[2]#	AJ16	311.123
DDR_CS[1]#	AK15	331.599
DDR_CS[0]#	AG16	308.11

Signal	Ball #	L_{PKG} (mils)
DDR_CMDCLKp[3]	AR23	778.386
DDR_CMDCLKp[2]	AR25	778.678
DDR_CMDCLKp[1]	AJ21	778.389
DDR_CMDCLKp[0]	AT22	779.052
DDR_CMDCLKn[3]	AR22	778.422
DDR_CMDCLKn[2]	AP25	778.626
DDR_CMDCLKn[1]	AH21	778.364
DDR_CMDCLKn[0]	AT23	778.663
DDR_DQSp[17]	AN23	647.885
DDR_DQSp[16]	AR13	662.658
DDR_DQSp[15]	AR14	756.37
DDR_DQSp[14]	AR18	657.177
DDR_DQSp[13]	AG18	883.127
DDR_DQSp[12]	AT29	803.514
DDR_DQSp[11]	AP26	570.199
DDR_DQSp[10]	AN35	954.634
DDR_DQSp[9]	AK29	743.421
DDR_DQSp[8]	AL22	649.037
DDR_DQSp[7]	AT11	664.122
DDR_DQSp[6]	AR12	751.289
DDR_DQSp[5]	AM18	655.037
DDR_DQSp[4]	AL18	883.794
DDR_DQSp[3]	AP27	799.988
DDR_DQSp[2]	AL25	570.205
DDR_DQSp[1]	AP34	954.578
DDR_DQSp[0]	AL30	743.195
DDR_DQSn[17]	AN22	649.015
DDR_DQSn[16]	AP13	663.855
DDR_DQSn[15]	AP14	753.405
DDR_DQSn[14]	AT18	656.553
DDR_DQSn[13]	AH18	884.398
DDR_DQSn[12]	AR29	798.671
DDR_DQSn[11]	AN26	567.625
DDR_DQSn[10]	AN34	953.297
DDR_DQSn[9]	AK30	743.793
DDR_DQSn[8]	AM22	648.058
DDR_DQSn[7]	AR11	666.771
DDR_DQSn[6]	AN12	750.783
DDR_DQSn[5]	AN18	657.838
DDR_DQSn[4]	AK18	884.327
DDR_DQSn[3]	AR28	802.131
DDR_DQSn[2]	AM25	570.284
DDR_DQSn[1]	AP35	954.857
DDR_DQSn[0]	AL29	741.936
DDR_CKE[3]	AK35	909.298

Signal	Ball #	L_{PKG} (mils)
DDR_CKE[2]	AJ26	418.229
DDR_CKE[1]	AL28	523.495
DDR_CKE[0]	AK32	576.501
PEA0_Tn[7]	AC29	593.556
PEA0_Tn[6]	AE29	620.958
PEA0_Tn[5]	AF31	576.632
PEA0_Tn[4]	AE35	700.292
PEA0_Tn[3]	AA29	596.186
PEA0_Tn[2]	Y35	645.822
PEA0_Tn[1]	AB34	620.014
PEA0_Tn[0]	AA33	655.968
PEA0_Tp[7]	AC30	593.725
PEA0_Tp[6]	AE30	621.066
PEA0_Tp[5]	AF32	576.466
PEA0_Tp[4]	AE36	698.793
PEA0_Tp[3]	AA30	596.151
PEA0_Tp[2]	Y36	645.75
PEA0_Tp[1]	AB35	619.949
PEA0_Tp[0]	AA34	656.085
PEA0_Rn[7]	AE32	537.172
PEA0_Rn[6]	AD28	620.151
PEA0_Rn[5]	AF34	655.661
PEA0_Rn[4]	AD33	578.785
PEA0_Rn[3]	AA26	614.031
PEA0_Rn[2]	Y32	674.826
PEA0_Rn[1]	AB30	584.459
PEA0_Rn[0]	Y28	615.49
PEA0_Rp[7]	AE33	537.665
PEA0_Rp[6]	AD29	620.238
PEA0_Rp[5]	AF35	655.816
PEA0_Rp[4]	AD34	578.475
PEA0_Rp[3]	AA27	613.895
PEA0_Rp[2]	Y33	674.933
PEA0_Rp[1]	AB31	584.59
PEA0_Rp[0]	Y29	615.353
PEA_CLKp	AC33	543.223
PEA_CLKn	AC32	543.066
PEB0_Tp[3]	L4	737.427
PEB0_Tp[2]	K3	738.08
PEB0_Tp[1]	H2	815.58
PEB0_Tp[0]	M6	601.538
PEB0_Tn[3]	L5	737.403
PEB0_Tn[2]	K2	738.047
PEB0_Tn[1]	H1	815.334
PEB0_Tn[0]	M7	601.008

Signal	Ball #	L_{PKG} (mils)
PEB0_Rp[3]	M3	601.653
PEB0_Rp[2]	K5	595.889
PEB0_Rp[1]	J4	689.625
PEB0_Rp[0]	L7	626.945
PEB0_Rn[3]	M4	601.876
PEB0_Rn[2]	K6	596.351
PEB0_Rn[1]	J3	689.585
PEB0_Rn[0]	L8	627.379
PEB_CLKp	M1	791.225
PEB_CLKn	L1	791.485
CLK14	B34	944.126
CLK48	H11	266.869
TEST#	AE28	327.716
DEBUG[7]	AG34	618.834
DEBUG[6]	AF29	390.112
DEBUG[5]	AH30	437.68
DEBUG[4]	AF27	286.798
DEBUG[3]	AG33	568.315
DEBUG[2]	AE27	268.077
DEBUG[1]	AG35	673.639
DEBUG[0]	AG31	489.947
USBp[3]	C7	721.092
USBp[2]	E8	626.27
USBp[1]	D9	570.561
USBp[0]	C10	570.597
USBn[3]	C6	720.726
USBn[2]	E7	626.63
USBn[1]	D8	570.488
USBn[0]	C9	570.542
SATA_CLKp	G24	713.291
SATA_CLKn	G23	713.128
SATA_RBIASp	J25	404.52
SATA_RBIASn	H25	453.275
SATA_TXp[5]	L19	199.904
SATA_TXp[4]	J21	320.709
SATA_TXp[3]	H19	339.094
SATA_TXp[2]	L16	120.841
SATA_TXp[1]	J15	228.172
SATA_TXp[0]	H13	299.933
SATA_TXn[5]	L20	199.463
SATA_TXn[4]	J22	321.021
SATA_TXn[3]	H20	340.013
SATA_TXn[2]	L17	120.25
SATA_TXn[1]	J16	227.708
SATA_TXn[0]	H14	299.979

Signal	Ball #	L_{PKG} (mils)
SATA_RXp[5]	H23	401.231
SATA_RXp[4]	K21	308.489
SATA_RXp[3]	G21	382.404
SATA_RXp[2]	K18	174.64
SATA_RXp[1]	H17	264.751
SATA_RXp[0]	K15	251.021
SATA_RXn[5]	H22	400.834
SATA_RXn[4]	K20	307.875
SATA_RXn[3]	G20	381.624
SATA_RXn[2]	K17	174.379
SATA_RXn[1]	H16	264.545
SATA_RXn[0]	K14	251.471
LAD[3]/FWH[3]	C19	514.663
LAD[2]/FWH[2]	B19	571.269
LAD[1]/FWH[1]	B20	536.694
LAD[0]/FWH[0]	A20	589.17
LFRAME#/FWH[4]	D20	419.248
UART_CLK	W30	348.997
SIU_RXD[2]	V30	357.176
SIU_RXD[1]	V27	270.552
SIU_TXD[2]	V33	486.628
SIU_TXD[1]	V35	593.658
SIU_CTS[2]#	W34	568.498
SIU_CTS[1]#	V36	663.999
SIU_DSR[2]#	W33	519.964
SIU_DSR[1]#	W35	615.097
SIU_DCD[2]#	V32	449.982
SIU_DCD[1]#	V34	534.175
SIU_RI[2]#	W28	271.018
SIU_RI[1]#	V29	315.662
SIU_DTR[2]#	W29	294.441
SIU_DTR[1]#	V26	288.551
SIU_RTS[2]#	W32	423.281
SIU_RTS[1]#	V31	386.175

11.2.1 Other Interface Signal Package Trace Length Data

If signals are not listed in Table 130, package trace length compensation is not required.Because package trace lengths are required for simulation, they are all documented in the signal integrity models.

12.0 Supported DRAM Technology

12.1 Memory Interface

The Intel® 3100 Chipset integrates a memory controller for direct connection to one channel of registered ECC DDR2-400 memory (single or dual ranked) with a maximum of up to 4 ranks. Peak theoretical memory data bandwidth using DDR2-400 technology is 3.2 GB/s. The minimum and maximum supported DDR2-400 memory configurations are listed in Table 131. ECC support allows for standard SEC-DED ECC. The Intel® 3100 Chipset supports registered ECC DIMMs.

Table 131. DDR2-400 Memory Interface Capacities

Processor FSB Address size	512 Mbit		1 Gbit		2 Gbit	
	Min	Max	Min	Max	Min	Max
32-bit	512 MB	4096 MB	1024 MB	4096 MB	2048 MB	4096 MB
36-bit	512 MB	4096 MB	1024 MB	8192 MB	2048 MB	16384 MB

12.2 Memory Interface Performance Optimizations

12.2.1 DDR2 Overlapped Command Scheduling

The memory controller command scheduler overlaps scheduling of activate and precharge commands for successive accesses, thus hiding the latency of these events. An activate command may be issued prior to the current command completion depending upon the bank addressed. It may be issued during the Trcd or Tcl wait periods Table 181 (Trcd bits 11:10 and Tcl bits 3:2). Overlap Command Scheduling is enabled in the DRC register.

12.2.2 Aggressive Page-Closed Policy with Look-Ahead

The Intel® 3100 Chipset uses an aggressive page closing policy with a single entry look-ahead. All commands are issued with auto-precharge unless a look-ahead indicates that a page-hit is already scheduled. The normal operating case is “page-empty”. The “page-hit” case will occur as often as the inbound/outbound and memory control arbiters are able to forward sequential requests from the same source back to back to the DDR2 interface, making this the next most frequent occurrence. A “page-miss” case is rare in the Intel® 3100 Chipset. Page closing policies are adjusted in the DRT register Table 181, “Offset 78 - 7Bh: DRT – DRAM Timing Register” on page 315.

12.3 Memory Address Translation Tables

Figure 73 defines the address bit translation from the system address to the DRAM row/column/bank address. Note that the count of DRAM devices per DIMM listed in the following table excludes the devices utilized for ECC information, thus the actual DRAM counts in the Intel® 3100 Chipset platforms is expected to be nine devices per rank of x8 technology, and 18 devices per rank of x4 technology.

Figure 73. Memory Channel Address Map

Row Size / Page Size	R/C/B		BA2	BA1	BA0	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
512 MB	14 x 10 x 2	Row	"0"	9	8	"0"	27	26	25	24	23	22	21	20	19	18	17	16	15	14
1KB/ Device		Col				"0"	"0"	"0"	"0"	AP	28	13	12	11	10	7	6	5	"0"	"0"
1024 MB	14 x 10 x 3	Row	29	9	8	"0"	27	26	25	24	23	22	21	20	19	18	17	16	15	14
1KB/ Device		Col				"0"	"0"	"0"	"0"	AP	28	13	12	11	10	7	6	5	"0"	"0"
2048MB	15 x 10 x 3	Row	30	9	8	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14
1KB/ Device		Col				"0"	"0"	"0"	"0"	AP	29	13	12	11	10	7	6	5	"0"	"0"

12.4 DIMM Topologies

The Intel® 3100 Chipset supports registered DDR2-400 DIMM technology, and is verified by Intel with registered ECC DIMMs only.Table 132 summarizes the supported DIMM populations.

Table 132. Supported DIMM Populations

Data Rate (MT/s)	Registered ECC DIMMs per channel	Technology (Device width)	Device Densities	Max Capacity
400	1-4 single-rank 1-2 dual-rank	x4, x8	512 Mb, 1 Gb, 2Gb	16 GB[1]
1. Max capacity is also limited by the processor's FSB address size.				

The number of supported ranks (electrical loads) per channel is limited by signal integrity considerations. The effect of these limitations varies with technology family (DDR2), and with data rate. Loading considerations also limit the viable mix of populated DIMM types, as their positions relative to the Intel® 3100 Chipset and to each other impact the signaling environment. In general, population of available DIMM slots must progress from the slot furthest from the Intel® 3100 Chipset towards the nearest.

12.4.1 DDR2-400 DIMM Slot Populations

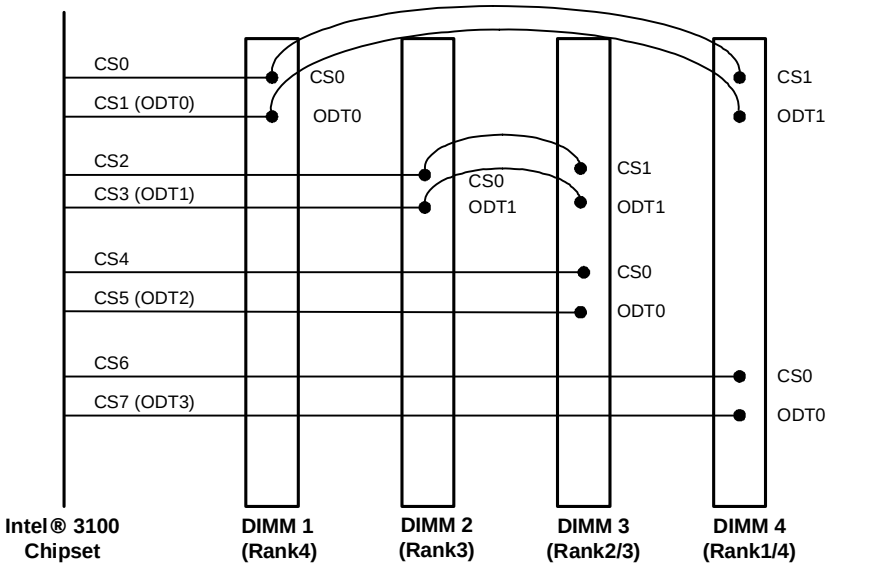
Support for DDR2-400 is limited to a maximum of four ranks, and is only verified by Intel in the On-Die Termination (ODT) enabled mode of operation. ODT support is provided via multi-mode operation on the "odd" chip-select pins of the channel, where

each odd CS runs the ODT protocol associated with the preceding even CS. (i.e., CS1 runs the ODT protocol associated with CS0, CS3 for CS2, CS5 for CS4, and CS7 for CS6.)

It is possible to construct a memory topology for DDR2 that supports both single and dual-rank DDR2 DIMM technology, provided the total number of populated ranks remains less than or equal to four. The dual-rank capable topology verified and supported by Intel is illustrated in Figure 74.

Note: The CS assignments are "standard" for single-rank population, and become less intuitive for dual-rank population. BIOS must be aware of a board routed in this fashion to generate the right DRM register setting.

Figure 74. DDR2 CS Routing for Single- or Dual-rank DIMMS



Supported DDR2-400 DIMM populations assuming the routing topology illustrated in Figure 74 are detailed in Table 133. Reference to "E" in the table signifies an empty rank.

Table 133. Supported DDR2-400 DIMM Populations (Sheet 1 of 2)

DRM = 1248h	DIMM 1		DIMM 2		DIMM 3		DIMM 4	
	Rank0	Rank1	Rank0	Rank1	Rank0	Rank1	Rank0	Rank1
1 single-rank	E	E	E	E	E	E	DRB0 CS6	E
1 dual-rank	E	E	E	E	E	E	DRB0 CS6	DRB6 CS0
2 single-rank	E	E	E	E	DRB4 CS4	E	DRB0 CS6	E
1 single / 1 dual	E	E	E	E	DRB4 CS4	E	DRB0 CS6	DRB6 CS0
2 dual-rank	E	E	E	E	DRB4 CS4	DRB2 CS2	DRB0 CS6	DRB6 CS0

Table 133. Supported DDR2-400 DIMM Populations (Sheet 2 of 2)

DRM = 1248h	DIMM 1		DIMM 2		DIMM 3		DIMM 4	
	Rank0	Rank1	Rank0	Rank1	Rank0	Rank1	Rank0	Rank1
3 single-rank	E	E	DRB2 CS2	E	DRB4 CS4	E	DRB0 CS6	E
2 single / 1 dual	E	E	DRB2 CS2	E	DRB4 CS4	E	DRB0 CS6	DRB6 CS0
4 single-rank	DRB6 CS0	E	DRB2 CS2	E	DRB4 CS4	E	DRB0 CS6	E

The DDR2ODTC register, Table 200 on page 338, must be programmed to select the desired ODT pin assertion combinations associated with each populated rank of DRAM. Since the setting of the DDR2ODTC register depends on the DRM register, it is both a logical CS and a logical ODT vector.

The recommended settings for the topology in Figure 74 are provided in Table 134. These values shown are for the logical ODT vector and the logical CS in the DDR2ODTC register. In the table below, E = empty, S = single and D = dual

Table 134. DDR2 ODTC Settings for DDR2

DRM (D0, F0, 80h) = 1248h	DRC (D0, F0, 7Ch)	SDRC (D0, F0, 88h)	DDR2 ODT Control Register (D0, F0, B0h) [0=off, 1=on]							
Topology	Bit 5	Bits 29:28	31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
Mapping	ODT Enable [0=Enable, 1=Disable]	Intel® 3100 Chipset ODT	Rank4 (CS0) ODT0 (CS1)		Rank3 (CS2) ODT1 (CS3)		Rank2 (CS4) ODT2 (CS5		Rank1 (CS6) ODT3 (CS7)	
			Wr	Rd	Wr	Rd	Wr	Rd	Wr	Rd
E-E-E-S	0	01	0000	0000	0000	0000	0000	0000	0001	0000
E-E-E-D	0	01	1001	0000	0000	0000	0000	0000	1001	0000
E-E-S-S	0	01	0000	0000	0000	0000	0011	0001	0011	0010
E-E-S-D	0	01	1010	0010	0000	0000	1011	1001	0011	0010
E-E-D-D	0	01	1110	0110	1101	1001	1011	1001	0111	0110
E-S-S-S	0	01	0000	0000	0111	0011	0101	0101	0110	0110
E-S-S-D	0	01	1110	0110	1101	1101	1101	1011	0111	0110
S-S-S-S	0	01	0111	0111	1011	1011	1101	1101	1110	1110

Note: The recommended values for DDR2 ODTC differ subtly for dual-rank versus single rank DIMM populations. The values provided offer superior signal integrity over more straightforward settings, and the BIOS developer is strongly encouraged to accommodate these values in a look-up table.

12.4.2 DQ/DQS Mapping

The data signal (DQ) to data strobe (DQS) relationship is controlled by the setting in the DRAM Row Attribute (DRA) registers, Table 176, "Offset 70h: DRA0 – DRAM Row 0 Attribute Register" on page 313. Bits 7:6 and 3:2 of these registers respectively define the device width for the odd and even rows, which is also used in the mapping of DQS signals to DQ signals.

Table 135. DRA Mapping for DQS

Bits	Definition	DQS per DQ
00	Reserved	NA
01	x8 DDR2	1 strobe per data byte
10	x4 DDR2	1 strobe per data nibble
11	Reserved	NA

Table 135 shows the mapping of DQS to DQ in general terms. Table 136 gives the exact relationship. x4 and x8 DDR2 mode use the same DQS to DQ mapping.

Note: DQ Nibble 0 consists of DQ bits 03:00, etc.

Table 136. DQS to DQ Mapping

DQ Nibble	DQS	
	x4 (Nibble)	x8 (byte)
0	0	0
1	9	0
2	1	1
3	10	1
4	2	2
5	11	2
6	3	3
7	12	3
8	4	4
9	13	4
10	5	5
11	14	5
12	6	6
13	15	6
14	7	7
15	16	7
16	8	8
17	17	8

12.5 Interface Signaling Voltage

The Intel® 3100 Chipset supports 1.8 V signaling for DDR2-400 DIMMs, based on the voltage supplied by the platform for the DDR2.

12.6 Clock Gearing Ratios

Clock gearing is a method of transferring data between clock domains when those clock domains are multiples of a common frequency, and the relationship between these two clocks is known. The matrix in Table 137 shows the FSB frequencies versus the DDR2 frequencies with the corresponding gearing ratio between the FSB base clock frequency and DDR2 base clock frequency. Note that data transfers are quad-pumped on the FSB and double-pumped on DDR2.

Table 137. Supported DDR2/FSB Clock Gearing Ratios

DDR2 Interface Clock Gearing Ratio		Memory Interface Technology and data rate
		DDR2 400
FSB Frequency (base/data)	100/400	2:1
	133/533	3:2
	167/667	6:5

12.7 Supported DRAM Timings

The Intel® 3100 Chipset is highly configurable in its DRAM timing configuration, but only a limited subset of the setting combinations possible are verified by Intel. The approved and expected settings for the various flavors of supported memory are listed in Table 138. For more information on the DRT register see Table 181, "Offset 78 - 7Bh: DRT – DRAM Timing Register" on page 315.

Table 138. Supported DRAM Timings

Type (T_{cl}-T_{rcd}-T_{rp}), (RdPtr Delay)	DRT value (HEX)	t_{CL} CAS latency	t_{RCD} RAS-CAS delay	t_{RP} RAS precharge
DDR2-400 (3-4-4), (1)	555E 9644	3	4	4
DDR2-400 (3-3-3), (1)	555E 5144	3	3	3
DDR2-400 (4-4-4), (1)	555E 9648	4	4	4
Note: This information applies to 512 Mb density devices. Refer to the DRT register description for device density dependent changes in the Trfc parameter.				

For the DRT values above, the internal read pointer delay setting is "01" binary for DDR2-400; indicating one additional clock. The corresponding delays are about 10ns for DDR2-400. The total timing budget is made up of both flight time and set-up requirements. A setting of "00" allows for a pointer delay of about ½ cycle, which only covers the internal set-up requirements. Flight times can vary between about 6-9 ns. This number depends on the topology (includes number of slots) and the loading (e.g., single/dual rank devices). Platforms with routing topologies that can support a lower read pointer delay setting (due to lower flight times) will save one DDR2 clock on all memory access latencies.

12.7.1 On-Die Termination (ODT)

The *JEDEC DDR2 DRAM Specification* requires that DDR2 DIMM devices provide selectable On-Die Termination (ODT) as an alternative to traditional discrete termination on the motherboard. The ODT feature is enabled via n extended memory register select EMRS command ODTENA (see Section 186, "Offset 88 - 8Bh: SDRC – Secondary DRAM Controller Mode Register" on page 326, bits 29 and 28) and supports

both 75 and 150 terminations activated dynamically via dedicated ODT interface signals on the DIMM. The Intel® 3100 Chipset supports operation of DDR2 devices with or without on-Die termination ODTenabled.

The Intel® 3100 Chipset utilizes its odd chip-select outputs to control the ODT pins of the DDR2 DIMM slot. Refer to Figure 74 for detail on routing of CS lines in ODT mode. Further detail on ODT configuration and operation across supported DIMM populations and topologies is provided in Table 134.

Note:

13.0 IMCH Registers

13.1 Device 0, Function 0: IMCH Registers

The Intel® 3100 Chipset Memory Controller Hub (IMCH) registers are in Device 0, Function 0. Table 139 provides the register address map for this device and function.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 139. IMCH Controller PCI Configuration Register Map (D0, F0) (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	01h	VID	Vendor Identification Register	8086h	RO
02h	03h	DID	Device Identification Register	35B0h	RO
04h	05h	PCICMD	PCI Command Register	0006h	RO, RW
06h	07h	PCISTS	PCI Status Register	0010h	RO, RWC
08h	08h	RID	Revision Identification Register	00h	RO
0Ah	0Ah	SUBC	Sub-Class Code Register	00h	RO
0Bh	0Bh	BCC	Base Class Code Register	06h	RO
0Eh	0Eh	HDR	Header Type Register	80h	RO
14h	17h	SMRBASE	System Memory RCOMP Base Address Register	0000_0000h	RO, RW
2Ch	2Dh	SVID	Subsystem Vendor Identification Register	0000h	RWO
2Eh	2Fh	SID	Subsystem Identification Register	0000h	RWO
4Ch	4Fh	NSIBAR	Root Complex Register Block Address Register	0000_0000h	RO, RW
50h	50h	IMCHCFG0	IMCH Configuration 0 Register	04h	RO, RW
51h	51h	IMCHCFG1	IMCH Configuration 1 Register	00000h	RO, RW
52h	52h	IMCHCFGNS0	IMCH Configuration 0 Register	00h	RO, RW
53h	53h	IMCHCFGNS1	IMCH Configuration 1 Register	00h	RO, RW0C
58h	58h	FDHC	Fixed DRAM Hole Control Register	00h	RO, RW
59h	59h	PAM0	Programmable Attribute Map 0 Register	00h	RO, RW
5Ah	5Ah	PAM1	Programmable Attribute Map 1 Register	00h	RO, RW
5Bh	5Bh	PAM2	Programmable Attribute Map 2 Register	00h	RO, RW
5Ch	5Ch	PAM3	Programmable Attribute Map 3 Register	00h	RO, RW
5Dh	5Dh	PAM4	Programmable Attribute Map 4 Register	00h	RO, RW
5Eh	5Eh	PAM5	Programmable Attribute Map 5 Register	00h	RO, RW
5Fh	5Fh	PAM6	Programmable Attribute Map 6 Register	00h	RO, RW
60h	60h	DRB0	DRAM Row 0 Boundary Register	00h	RW
61h	61h	DRB1	DRAM Row 1 Boundary Register	00h	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 139. IMCH Controller PCI Configuration Register Map (D0, F0) (Sheet 2 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
62h	62h	DRB2	DRAM Row 2 Boundary Register	00h	RW
63h	63h	DRB3	DRAM Row 3 Boundary Register	00h	RW
64h	64h	DRB4	DRAM Row 4 Boundary Register	00h	RW
65h	65h	DRB5	DRAM Row 5 Boundary Register	00h	RW
66h	66h	DRB6	DRAM Row 6 Boundary Register	00h	RW
67h	67h	DRB7	DRAM Row 7 Boundary Register	00h	RW
70h	70h	DRA0	DRAM Row 0 Attribute Register	00h	RW
71h	71h	DRA1	DRAM Row 1 Attribute Register	00h	RW
72h	72h	DRA2	DRAM Row 2 Attribute Register	00h	RW
73h	73h	DRA3	DRAM Row 3 Attribute Register	00h	RW
78h	7Bh	DRT	DRAM Timing Register	859A_9604h	RW
7Ch	7Fh	DRC	DRAM Controller Mode Register	0000_0000h	RO, RW, RWO
80h	81h	DRM	DRAM Mapping Register	8421h	RW
82h	82h	DRORC	Opportunistic Refresh Control Register	71h	RW
84h	87h	ECCDIAG	ECC Detection/Correction Diagnostic Register	0000_0000h	RO, RW, RWS
88h	8Bh	SDRC	DDR SDRAM Secondary Control Register	0000_0000h	RO, RW
8Ch	8Ch	CKDIS	CK/CK# Clock Disable Register	FFh	RW
8Dh	8Dh	CKEDIS	CKE/CKE# Clock Disable Register	00h	RW
90h	93h	SPARECTL	Spare Control Register	0000_0000h	RO, RW
94h	97h	DRAMISCTL	DRAM Miscellaneous Control Register	B030_0000h	RO, RW
9Ah	9Bh	DDRCSR	DDR Channel Configuration Control/Status Register	0000h	RO, RW, RWS
9Ch	9Ch	DEVPRES	Device Present Register	03h	RO, RWO
9Dh	9Dh	EXSMRC	Extended System Management RAM Control Register	00h	RO, RWL, RWC
9Eh	9Eh	SMRAM	System Management RAM Control Register	02h	RO, RW, RWC, RWS, RWL
9Fh	9Fh	EXSMRAMC	Expansion System Management RAM Control Register	07h	RO, RWC
A0h	A3h	CLKGRFM0	Clock Gearing Ratio FSB to Memory 0 Register	0015_4320h	RW
A4h	A7h	CLKGRFM1	Clock Gearing Ratio FSB to Memory 1 Register	0000_0000h	RW
A8h	ABh	CLKGRMF0	Clock Gearing Ratio Memory to FSB 0 Register	0006_5432h	RW
ACh	AFh	CLKGRMF1	Clock Gearing Ratio Memory to FSB 1 Register	0001_0000h	RW
B0h	B3h	DDR2ODTC	DDR2 ODT Control Register	0000_0000h	RW
C4h	C5h	TOLM	Top of Low Memory Register	0800h	RO, RW
C6h	C7h	REMAPBASE	Remap Base Address Register	03FFh	RO, RW
C8h	C9h	REMAPLIMIT	Remap Limit Address Register	0000h	RO, RW
CAh	CBh	REMAPOFFSET	Remap Offset Register	0000h	RO, RW
CCh	CDh	TOM	Top of Memory Register	0000h	RO, RW
CEh	CFh	HECBASE	PCI Express Port A (PEA) Enhanced Configuration Base Address Register	E000h	RO, RWO
DEh	DFh	SKPD	Scratchpad Data Register	0000h	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

13.1.1 Register Details

13.1.1.1 Offset 00 - 01h: VID – Vendor Identification Register

The VID Register contains the vendor identification number. This 16-bit register combined with the Device Identification Register uniquely identifies any PCI device. Writes to this register have no effect.

Table 140. Offset 00 - 01h: VID – Vendor Identification Register

Device: 0 ***Function:*** 0
Offset: 00 - 01h ***Size:*** 16 bit
Default Value: 8086h

Bits	Name	Description	Reset Value	Access
15:00	VID	**Vendor Identification:** This register field contains the PCI standard identification for Intel 8086h.	8086h	RO

13.1.1.2 Offset 02 - 03h: DID – Device Identification Register

This 16-bit register combined with the Vendor Identification register uniquely identifies any PCI device. Writes to this register have no effect.

Table 141. Offset 02 - 03h: DID – Device Identification Register

Device: 0 ***Function:*** 0
Offset: 02 - 03h ***Size:*** 16 bit
Default Value: 35B0h

Bits	Name	Description	Reset Value	Access
15:00	DID	**Device Identification Number:** This is a 16-bit value assigned to the IMCH Host-NSI Bridge Function 0.	35B0	RO

13.1.1.3 Offset 04 - 05h: PCICMD – PCI Command Register

Since IMCH Device 0 does not physically reside on a PCI bus, many of the bits are not supported.

Table 142. Offset 04 - 05h: PCICMD – PCI Command Register (Sheet 1 of 2)

Device: 0 ***Function:*** 0
Offset: 04 - 05h ***Size:*** 16 bit
Default Value: 0006h

Bits	Name	Description	Reset Value	Access
15:10	Reserved	Reserved	00h	
09	FB2B	**Fast Back-to-Back Enable:** This bit is hardwired to 0.	0b	RO

Table 142. Offset 04 - 05h: PCICMD – PCI Command Register (Sheet 2 of 2)

Device: 0	***Function:*** 0
Offset: 04 - 05h	***Size:*** 16 bit
Default Value: 0006h	

Bits	Name	Description	Reset Value	Access
08	SERRE	**SERR Enable:** This bit is a global enable bit for Device 0 SERR messaging. The IMCH does not have a SERR signal. The IMCH communicates the SERR condition by sending a SERR message over NSI to the IICH. 0 = Disable. The SERR message is not generated by the IMCH for Device 0. 1 = Enable. The IMCH enables generation of SERR messages over NSI for specific Device 0, Function 0 error conditions that are enabled via the PCICMD register. The error status is reported in the PCISTS registers. The only error event enabled through Device 0, Function 0 is Detected Parity Error which is essentially a NSI poisoned TLP, and is enabled by the parity error enable bit (PERRE). ***Note:*** This bit only controls SERR messaging for Device 0, Function 0. Device 0, Function 1, and Devices 1-7 have their own SERR bits to control error reporting for error conditions occurring on their respective devices. The control bits are used in a logical OR manner to enable the SERR NSI message mechanism.	0b	RW
07	Reserved	Reserved	0b	
06	PERRE	**Parity Error Enable:** 0 = Disable. The IMCH does not take any action when it detects data corruption on NSI. 1 = Enable. The IMCH generates an SERR message over the NSI to the IICH when a poisoned TLP is detected by the IMCH on NSI (DPE set in PCISTS) and SERRE is set to 1.	0b	RW
05:03	Reserved	Reserved	0h	
02	BME	**Bus Master Enable:** The IMCH is always enabled as a master on NSI. This bit is hardwired to 1. Writes to this bit position have no effect.	1b	RO
01	MAE	**Memory Access Enable:** This bit is hardwired to 1.	1b	RO
00	Reserved	Reserved	0b	

13.1.1.4 Offset 06 - 07h: PCISTS – PCI Status Register

PCISTS is a 16-bit status register that reports the occurrence of error events on Device 0's PCI interface. Since IMCH Device 0 does not physically reside on a PCI bus many of the bits are not supported.

Table 143. Offset 06 - 07h: PCISTS – PCI Status Register

Device: 0 **Function:** 0
Offset: 06 - 07h **Size:** 16 bit
Default Value: 0010h

Bits	Name	Description	Reset Value	Access
15	DPE	**Detected Parity Error:** This bit is set to 1 whenever it receives a poisoned TLP regardless of the state of the parity error response bit. Software may clear this by writing a 1 to this bit.	0b	RWC
14	SSE	**Signaled System Error:** 0 = Software clears this bit by writing a 1 to the bit location. 1 = IMCH Device 0, Function 0 generates a SERR message over NSI for any enabled Device 0, Function 0 error condition. Device 0 error conditions are enabled in the PCICMD register. Device 0 error flags are read/reset from the PCISTS register. The only error that can be enabled to signal system error through Device 0, Function 0 is the detected parity error which is essentially a NSI poisoned TLP. Software may clear this by writing a 1 to this bit.	0b	RWC
13	RMAS	**Received Master Abort Status:** This bit is set if the **IMCH** generates a **NSI** request that receives a completion with unsupported request completion status. Software may clear this by writing a 1 to this bit.	0b	RWC
12	RTAS	**Received Target Abort Status:** Set to 1 by hardware if the IMCH generated a request that received a completion with Completer Abort status. Software clears this bit by writing a 1 to this bit location.	0b	RWC
11	STAS	**Signaled Target Abort Status:** The IMCH does not generate a Completer Abort on the NSI completion packet. This bit is hardwired to w10. Writes to this bit position have no effect.	0b	RO
10:09	Reserved	Reserved	00b	
08	DPD	**Master Data Parity Error Detected:** This bit is hardwired to 0.	0b	RWC
07	FB2B	**Fast Back-to-Back:** Reserved.	0b	
06:05	Reserved	Reserved	00b	
04	CLIST	**Capability List:** This bit is hardwired to 1 to indicate to the configuration software that this device/function implements a list of new capabilities. A list of new capabilities is accessed via register CAPPTR at configuration address offset 34h.	1b	RO
03:00	Reserved	Reserved	0h	

13.1.1.5 Offset 08h: RID – Revision Identification Register

This register contains the revision number of the IMCH Device 0. These bits are read-only and writes to this register have no effect.

Table 144. Offset 08h: RID – Revision Identification Register

Device: 0 **Function:** 0
Offset: 08h **Size:** 8 bit
Default Value: 00h

Bits	Name	Description	Reset Value	Access
07:00	RID	**Revision Identification Number:** This value indicates the revision identification number for the IMCH Device 0. 00h = A0 stepping	00h	RO

13.1.1.6 Offset 0Ah: SUBC – Sub-Class Code Register

Table 145. Offset 0Ah: SUBC – Sub-Class Code Register

Device: 0	**Function:** 0
Offset: 0Ah	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07:00	SUBC	**Sub-Class Code:** This value indicates the Sub Class Code into which the IMCH Device 0 falls. 00h = Host Bridge	00h	RO

13.1.1.7 Offset 0Bh: BCC – Base Class Code Register

Table 146. Offset 0Bh: BCC – Base Class Code Register

Device: 0	**Function:** 0
Offset: 0Bh	**Size:** 8 bit
Default Value: 06h	

Bits	Name	Description	Reset Value	Access
07:00	BASEC	**Base Class Code:** This value indicates the Base Class Code for the IMCH Device 0. 06h = Bridge device	06h	RO

13.1.1.8 Offset 0Dh: MLT – Master Latency Timer Register

Device 0 in the IMCH is not a PCI master so this register is not supported.

13.1.1.9 Offset 0Eh: HDR – Header Type Register

Table 147. Offset 0Dh: MLT – Master Latency Timer Register

Device: 0	**Function:** 0
Offset: 0Dh	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07:00	Reserved	Reserved	00h	

Table 148. Offset 0Eh: HDR – Header Type Register

Device: 0	**Function:** 0
Offset: 0Eh	**Size:** 8 bit
Default Value: 80h	

Bits	Name	Description	Reset Value	Access
07:00	HDR	**PCI Header:** The header type of the IMCH Device 0. 80h = multi-function device with standard header layout. Note: This register should return a 00h indicating a single function device, when both functions 1 and 2 are disabled.	80h	RO

13.1.1.10 Offset 14 - 17h: SMRBASE – System Memory RCOMP Base Address Register

The SMRBASE is a standard PCI Base Address register that is used to set the base of the Memory Mapped Registers used to control the System Memory I/O Buffer RCOMP.

Note: All accesses to these Memory Mapped Registers must be made as a single Dword (4 bytes) or less. Access must be aligned on a natural boundary.

Table 149. Offset 14 - 17h: SMRBASE – System Memory RCOMP Base Address Register

Device: 0	**Function:** 0
Offset: 14 - 17h	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
31:12	UPBITS	**Upper Programmable Base Address:** These bits are part of the SM MMR region, normally set by configuration software to locate the base address of the region.	00000h	RW, RO
11:04	LOWBITS	**Lower Bits:** These bits are hardwired to 0. This forces the size of the memory region to be 4 Kbyte.	00h	RO

Table 149. Offset 14 - 17h: SMRBASE – System Memory RCOMP Base Address Register

Device: 0 ***Function:*** 0
Offset: *14 - 17h* ***Size:*** *32 bit*
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
03	PF	**Prefetchable:** This bit is hardwired to 0 to indicate that the System Memory MMR region is NON-Prefetchable.	0b	RO
02:01	TYPE	**Addressing Type:** These bits determine addressing type and they are hardwired to 00 to indicate that address range defined by the upper bits of this register can be located anywhere in the 32-bit address space in order to comply with the PCI specification for base address registers.	00b	RO
00	MSPACE	**Memory Space Indicator:** Hardwired to 0 to identify the MMR range as a memory range as per the specification for PCI base address registers.	0b	RO

13.1.1.11 Offset 2C - 2Dh: SVID – Subsystem Vendor Identification Register

This value is used to identify the vendor of the subsystem.

Table 150. Offset 2C - 2Dh: SVID – Subsystem Vendor Identification Register

Device: 0 ***Function:*** 0
Offset: *2C - 2Dh* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:00	SUBVID	**Subsystem Vendor ID:** This field must be programmed during boot-up to indicate the vendor of the system board.	0000h	RWO

13.1.1.12 Offset 2E - 2Fh: SID – Subsystem Identification Register

This value is used to identify a particular subsystem.

Table 151. Offset 2E - 2Fh: SID – Subsystem Identification Register

Device: 0 ***Function:*** 0
Offset: *2E - 2Fh* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:00	SUBID	**Subsystem ID:** This field must be programmed during BIOS initialization.	0000h	RWO

13.1.1.13 Offset 4C - 4Fh: NSIBAR – Root Complex Block Address Register

This is the base address for the Root Complex memory-mapped configuration space. This window of addresses contains the Root Complex Register Block for the NSI hierarchy associated with the IMCH. There is no physical memory within this 4 Kbyte window that can be addressed. The 4 Kbyte reserved by this register does not alias to any PCI 2.3 compliant memory mapped space.

All accesses to these Memory Mapped Registers must be made as a single Dword (4 bytes) or less. Access must be aligned on a natural boundary.

Table 152. Offset 4C - 4Fh: NSIBAR – Root Complex Block Address Register

Device: 0	**Function:** 0
Offset: 4C - 4Fh	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
31:12	NSI_BA	NSI **Base Address:** The BIOS programs this register resulting in a base address for a 4 Kbyte block of contiguous memory address space. This register ensures that a naturally aligned 4 Kbyte space is allocated within total addressable memory space of 4 Gbyte. System Software uses this base address to program the NSI register set.	00000h	RW
11:00	Reserved	Hardwired to 0.	000h	

13.1.1.14 Offset 50h: IMCH CFG0 – IMCH Configuration 0 Register

MCHCFG consists of IMCH CFG1 in the upper 8 bits and IMCH CFG0 in the lower 8 bits.

Table 153. Offset 50h: IMCH CFG0 – IMCH Configuration 0 Register

Device: 0	**Function:** 0
Offset: 50h	**Size:** 8 bit
Default Value: 04h	

Bits	Name	Description	Reset Value	Access
07:03	Reserved	Reserved	000b	
02	IOQD	**In-Order Queue Depth:** This bit reflects the value sampled on HA[7]# on the de-assertion of the CPURST#. It indicates the depth of the processor bus in-order queue. 0 = HA[7]# has been sampled asserted (e.g., logic one, or electrical low). The depth of the IOQ is set to one (e.g., no pipelining on the processor bus). HA[7]# may be driven low during CPURST# by an external source. 1 = HA[7]# was sampled as deasserted (e.g. logic zero or electrical high). The depth of the processor bus in-order queue is configured to the maximum (e.g., 12).	1b	RO
01	Reserved	Reserved	0b	
00	Reserved	Reserved	0b	

13.1.1.15 Offset 51h: IMCH CFG1 – IMCH Configuration 1 Register

Table 154. Offset 51h: IMCH CFG1 – IMCH Configuration 1 Register

Device: 0	**Function:** 0
Offset: 51h	**Size:** 8 bit
Default Value: 00000h	

Bits	Name	Description	Reset Value	Access
07:05	NSG	**Number of Stop Grant Cycles:** Number of Stop Grant transactions expected on the FSB bus before a packet is sent to the IICH. This field is programmed by the BIOS after it has enumerated the processors and before it has enabled Stop Clock generation in the IICH. Once this field has been set, it must not be modified. Note that each enabled thread within each processor generates Stop Grant Acknowledge transactions. ***Note:*** This register is read/write and not write-once as in some implementations. Encoding Description 0 0 0 NSI Stop Grant generated after 1 FSB Stop Grant 0 0 1 NSI Stop Grant generated after 2 FSB Stop Grant All other encoding options are reserved.	000b	RW
04:00	Reserved	Reserved	00000b	

13.1.1.16 Offset 52h: IMCH CFGNS0 – IMCH Configuration 0 Register

This register contains control and status bit for the DRAM memory scrubber, and status bits for the DRAM thermal management feature.

Table 155. Offset 52h: IMCH CFGNS0 – IMCH Configuration 0 Register

Device: 0	**Function:** 0
Offset: 52h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07	Reserved	Reserved	0b	
06:05	SCC	**Scrub Completion Counter:** This field reflects scrub iterations. Upon the first scrub completion, this increments to 01. Subsequent scrub rollovers increment this value through values 10,11,00,01 in sequence.	00b	RO
04	Reserved	Reserved	0b	
03:02	SCRUBR	**Scrub Rate:** These two bits determine the scrub counter time. For DRAM auto-initialization, these bits should be programmed with 00. This setting implies initialization to the memory controller. 00 Reserved 01 Fast mode of 256 clocks 10 32 k clocks, normal operation 11 Reserved	00b	RW
01:00	SME	**Scrub Mode Enable:** When scrub mode is enabled, user must change SCRUBR (3:2) register from default mode (00) to 01 or 10. 00 Scrub Engine turned off 01 Reserved 10 Scrub Engine in Normal Scrub mode 11 Reserved	00b	RW

13.1.1.17 Offset 53h: IMCH CFGNS1 – IMCH Configuration 1 Register

This register contains IMCH control bits that should not be sticky.

Table 156. Offset 53h: IMCH CFGNS1 – IMCH Configuration 1 Register

Device: 0	***Function:*** 0
Offset: *53h*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:02	Reserved	Reserved	000b	
01	THWO	**Throttled-Write Occurred:** 0 = Writing a zero clears this bit. 1 = This bit is set by hardware when a write is throttled. This happens when the maximum allowed number of writes has been reached during a time-slice and there is at least one more write to be completed.	0b	RW0C
00	THRO	**Throttled-Read Occurred:** 0 = Writing a zero clears this bit. 1 = This bit is set by hardware when a read is throttled. This happens when the maximum allowed number of reads has been reached during a time-slice and there is at least one more read to be done.	0b	RW0C

13.1.1.18 Offset 58h: FDHC – Fixed DRAM Hole Control Register

This 8-bit register controls a fixed DRAM hole from 15–16 Mbytes.

Table 157. Offset 58h: FDHC – Fixed DRAM Hole Control Register

Device: 0	***Function:*** 0
Offset: *58h*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07	HEN	**Hole Enable:** This field enables a memory hole in DRAM space. The DRAM that lies "behind" this space is not remapped. 0 = No memory hole 1 = Memory hole from 15–16 Mbytes. Accesses in this range are sent to NSI.	0b	RW
06:00	Reserved	Reserved	00h	

13.1.1.19 Offset 59h: PAM0 – Programmable Attribute Map 0 Register

This register controls the read, write, and shadowing attributes of the BIOS area from 0F0000h-0FFFFFh. See Section 5.1.3, "PAM Memory Spaces" for more information on PAM memory spaces.

Table 158. Offset 59h: PAM0 – Programmable Attribute Map 0 Register

Device: 0 **Function:** 0
Offset: 59h **Size:** 8 bit
Default Value: 00h

Bits	Name	Description	Reset Value	Access
07:06	Reserved	Reserved	00b	
05:04	HIENABLE	**Attribute Register:** This field controls the steering of read and write cycles that address the BIOS area from 0F0000 to 0FFFFF. Encoding Description: 0 0 DRAM Disabled - All accesses are directed to NSI 0 1 Read-Only - All reads are serviced by DRAM. All writes are forwarded to NSI 1 0 Write-Only - All writes are sent to DRAM. Reads are serviced by NSI 1 1 Normal DRAM Operation - All reads and writes are serviced by DRAM	00b	RW
03:00	Reserved	Reserved	0h	

13.1.1.20 Offset 5Ah: PAM1 – Programmable Attribute Map 1 Register

This register controls the read, write, and shadowing attributes of the BIOS areas from 0C0000h-0C7FFFh.

Table 159. Offset 5Ah: PAM1 – Programmable Attribute Map 1 Register

Device: 0 **Function:** 0
Offset: 5Ah **Size:** 8 bit
Default Value: 00h

Bits	Name	Description	Reset Value	Access
07:06	Reserved	Reserved	00b	
05:04	HIENABLE	**Attribute Register 0C4000-0C7FFF:** This field controls the steering of read and write cycles that address the BIOS area from 0C4000 to 0C7FFF Encoding Description 0 0 DRAM Disabled - All accesses are directed to NSI 0 1 Read-Only - All reads are serviced by DRAM. All writes are forwarded to NSI 1 0 Write-Only - All writes are sent to DRAM. Reads are serviced by NSI 1 1 Normal DRAM Operation - All reads and writes are serviced by DRAM	00b	RW
03:02	Reserved	Reserved	00b	
01:00	LOENABLE	**Attribute Register 0C0000-0C3FFF:** This field controls the steering of read and write cycles that address the BIOS area from 0C0000 to 0C3FFF. Encoding Description 0 0 DRAM Disabled - All accesses are directed to NSI 0 1 Read-Only - All reads are serviced by DRAM. All writes are forwarded to NSI 1 0 Write-Only - All writes are sent to DRAM. Reads are serviced by NSI 1 1 Normal DRAM Operation - All reads and writes are serviced by DRAM	00b	RW

13.1.1.21 Offset 5Bh: PAM2 – Programmable Attribute Map 2 Register

This register controls the read, write, and shadowing attributes of the BIOS areas from 0C8000h-0CFFFFh.

Table 160. Offset 5Bh: PAM2 – Programmable Attribute Map 2 Register

Device: 0 ***Function:*** 0
Offset: 5Bh ***Size:*** *8 bit*
Default Value: *00h*

Bits	Name	Description	Reset Value	Access
07:06	Reserved	Reserved	00b	
05:04	HIENABLE	**Attribute Register 0CC000-0CFFFF:** Encoding Description 0 0 DRAM Disabled - All accesses are directed to NSI 0 1 Read-Only - All reads are serviced by DRAM. All writes are forwarded to NSI 1 0 Write-Only - All writes are sent to DRAM. Reads are serviced by NSI 1 1 Normal DRAM Operation - All reads and writes are serviced by DRAM	00b	RW
03:02	Reserved	Reserved	00b	
01:00	LOENABLE	**Attribute Register 0C8000-0CBFFF:** This field controls the steering of read and write cycles that address the BIOS area from 0C8000 to 0CBFFF. Encoding Description 0 0 DRAM Disabled - All accesses are directed to NSI 0 1 Read-Only - All reads are serviced by DRAM. All writes are forwarded to NSI 1 0 Write-Only - All writes are sent to DRAM. Reads are serviced by NSI 1 1 Normal DRAM Operation - All reads and writes are serviced by DRAM	00b	RW

13.1.1.22 Offset 5Ch: PAM3 – Programmable Attribute Map 3 Register

This register controls the read, write, and shadowing attributes of the BIOS areas from 0D0000h-0D7FFFh.

Table 161. Offset 5Ch: PAM3 – Programmable Attribute Map 3 Register

Device: 0	**Function:** 0
Offset: 5Ch	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07:06	Reserved	Reserved	00b	
05:04	HIENABLE	**Attribute Register 0D4000-0D7FFF:** This field controls the steering of read and write cycles that address the BIOS area from 0D4000 to 0D7FFF. Encoding Description 0 0 DRAM Disabled - All accesses are directed to NSI 0 1 Read-Only - All reads are serviced by DRAM. All writes are forwarded to NSI 1 0 Write Only - All writes are sent to DRAM. Reads are serviced by NSI 1 1 Normal DRAM Operation - All reads and writes are serviced by DRAM	00b	RW
03:02	Reserved	Reserved	00b	
01:00	LOENABLE	**Attribute Register 0D0000-0D3FFF:** This field controls the steering of read and write cycles that address the BIOS area from 0D0000 to 0D3FFF. Encoding Description 0 0 DRAM Disabled - All accesses are directed to NSI 0 1 Read-Only - All reads are serviced by DRAM. All writes are forwarded to NSI 1 0 Write Only - All writes are sent to DRAM. Reads are serviced by NSI 1 1 Normal DRAM Operation - All reads and writes are serviced by DRAM	00b	RW

13.1.1.23 Offset 5Dh: PAM4 – Programmable Attribute Map 4 Register

This register controls the read, write, and shadowing attributes of the BIOS areas from 0D8000h-0DFFFFh.

Table 162. Offset 5Dh: PAM4 – Programmable Attribute Map 4 Register

Device: 0	**Function:** 0
Offset: 5Dh	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07:06	Reserved	Reserved	00b	

Table 162. Offset 5Dh: PAM4 – Programmable Attribute Map 4 Register

Device: 0 **Function:** 0
Offset: 5Dh **Size:** 8 bit
Default Value: 00h

Bits	Name	Description	Reset Value	Access
05:04	HIENABLE	**Attribute Register 0DC000-0DFFFF:** This field controls the steering of read and write cycles that address the BIOS area from 0DC000 to 0DFFFF. Encoding Description 0 0 DRAM Disabled - All accesses are directed to NSI 0 1 Read-Only - All reads are serviced by DRAM. All writes are forwarded to NSI 1 0 Write Only - All writes are sent to DRAM. Reads are serviced by NSI 1 1 Normal DRAM Operation - All reads and writes are serviced by DRAM	00b	RW
03:02	Reserved	Reserved	00b	
01:00	LOENABLE	**Attribute Register 0D8000-0DBFFF:** This field controls the steering of read and write cycles that address the BIOS area from 0D8000 to 0DBFFF. Encoding Description 0 0 DRAM Disabled - All accesses are directed to NSI 0 1 Read-Only - All reads are serviced by DRAM. All writes are forwarded to NSI 1 0 Write Only - All writes are sent to DRAM. Reads are serviced by NSI 1 1 Normal DRAM Operation - All reads and writes are serviced by DRAM	00b	RW

13.1.1.24 Offset 5Eh: PAM5 – Programmable Attribute Map 5 Register

This register controls the read, write, and shadowing attributes of the BIOS areas from 0E0000h-0E7FFFh.

Table 163. Offset 5Eh: PAM5 – Programmable Attribute Map 5 Register

Device: 0	***Function:*** 0
Offset: *5Eh*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:06	Reserved	Reserved	00b	
05:04	HIENABLE	**Attribute Register 0E4000-0E7FFF:** This field controls the steering of read and write cycles that address the BIOS area from 0E4000 to 0E7FFF. Encoding Description 0 0 DRAM Disabled - All accesses are directed to NSI 0 1 Read-Only - All reads are serviced by DRAM. All writes are forwarded to NSI 1 0 Write Only - All writes are sent to DRAM. Reads are serviced by NSI 1 1 Normal DRAM Operation - All reads and writes are serviced by DRAM	00b	RW
03:02	Reserved	Reserved	00b	
01:00	LOENABLE	**Attribute Register 0E0000-0E3FFF:** This field controls the steering of read and write cycles that address the BIOS area from 0E0000 to 0E3FFF. Encoding Description 0 0 DRAM Disabled - All accesses are directed to NSI 0 1 Read-Only - All reads are serviced by DRAM. All writes are forwarded to NSI 1 0 Write Only - All writes are sent to DRAM. Reads are serviced by NSI 1 1 Normal DRAM Operation - All reads and writes are serviced by DRAM	00b	RW

13.1.1.25 Offset 5Fh: PAM6 – Programmable Attribute Map 6 Register

This register controls the read, write, and shadowing attributes of the BIOS areas from 0E8000h-0EFFFFh.

Table 164. Offset 5Fh: PAM6 – Programmable Attribute Map 6 Register

Device: 0 ***Function:*** 0
Offset: *5Fh* ***Size:*** *8 bit*
Default Value: *00h*

Bits	Name	Description	Reset Value	Access
07:06	Reserved	Reserved	00b	
05:04	HIENABLE	**Attribute Register 0EC000-0EFFFF:** This field controls the steering of read and write cycles that address the BIOS area from 0EC000 to 0EFFFF. Encoding Description 0 0 DRAM Disabled - All accesses are directed to NSI 0 1 Read-Only - All reads are serviced by DRAM. All writes are forwarded to NSI 1 0 Write Only - All writes are sent to DRAM. Reads are serviced by NSI 1 1 Normal DRAM Operation - All reads and writes are serviced by DRAM	00b	RW
03:02	Reserved	Reserved	00b	
01:00	LOENABLE	**Attribute Register 0E8000-0EBFFF:** This field controls the steering of read and write cycles that address the BIOS area from 0E8000 to 0EBFFF. Encoding Description 0 0 DRAM Disabled - All accesses are directed to NSI 0 1 Read-Only - All reads are serviced by DRAM. All writes are forwarded to NSI 1 0 Write Only - All writes are sent to DRAM. Reads are serviced by NSI 1 1 Normal DRAM Operation - All reads and writes are serviced by DRAM	00b	RW

13.1.1.26 Offset 60h: DRB0 – DRAM Row 0 Boundary Register

The DRAM Row Boundary (DRB) register defines the upper boundary address of each DRAM row with a granularity of 64 Mbyte as reflected in CSR (Table 191). Each row has its own single-byte DRB register. The value in a given DRB corresponds to the cumulative memory size including that row. For example, a value of 1 (000 0001) in DRB0 (address lines 33 to 26) indicates that 64 Mbytes of DRAM has been populated in the first row.

Table 165. Offset 60h: DRB0 – DRAM Row 0 Boundary Register

Device: 0 ***Function:*** 0
Offset: *60h* ***Size:*** *8 bit*
Default Value: *00h*

Bits	Name	Description	Reset Value	Access
07:00	DRAM_RBA	**DRAM Row Boundary Address:** This 8 bit value defines the upper address for each row of DRAM rows. This 8 bit value is compared against a set of address lines to determine the upper address limit of a particular row. This field corresponds to bits 33:26 of the system address.	00h	RW

Table 166. DRB to DIMM Designation

	Even Row (or Single Rank)		Odd Row (Present if Dual Rank)	
	Row Number	Address of DRB	Row Number	Address of DRB
DIMM 1	Row 0	60h	Row 1	61h
DIMM 2	Row 2	62h	Row 3	63h
DIMM 3	Row 4	64h	Row 5	65h
DIMM 4	Row 6	66h	Row 7	67h

DRB0 = Total memory in row0 (64 Mbyte increments)

DRB1 = Total memory in row0 + row1 (64 Mbyte increments)

DRB7 = Total populated memory in row0 + row1 + row2 + row3 + row4 + row5 + row6 + row7 (in 64 Mbyte increments)

The functionality of DRB7 is somewhat different from that of DRB[6:0]. If DRB(6:0) are non-zero and DRB7 is set to a value of 00h after memory configuration, the value is interpreted as 100h. This functionality avoids a 64 Mbyte “hole” at the top of memory. This behavior is unique to the DRB7 register.

The row referred to by this register is defined by the DIMM chip select used. For DDR2, only logical single rank devices are supported, even though the devices themselves might be physically dual rank. This affects how the DRBs are programmed. Dual-rank DIMM’s use both Row0 and Row2 (for CS0# and CS2#), even though there is one physical slot for the row, because they are treated as 2 logical single ranks. Single rank DIMM’s use only the even row number, since single rank DIMM’s only support CS0#. For single-rank DIMM’s the value BIOS places in the odd row must equal the same value as what was placed in the even row field. A row is 64b wide interface consisting of one DIMM.

Unpopulated rows must be programmed with the value of the last populated row.

ODT support is provided via multi-mode operation on the “odd” chip-select pins of each channel, where each odd CS will run the ODT protocol associated with the preceding even CS. (i.e.: CS1# runs the ODT protocol associated with CS0#, CS3# for CS2#, CS5# for CS4#, and CS7# for CS6#.)

Programming Example:

Configuration for 64 Mbyte granularity with DIMMs populated as shown in Table 167.

Table 167. Example Configuration

DIMM1	512 Mbyte in even row, none in odd row (single rank DIMM)
DIMM2	1 Gbyte in even and odd rows (dual rank DIMM)
DIMM3	1 Gybte in even row, none in odd row (single rank DIMM)
DIMM4	2 Gbyte in even row and odd rows (dual rank DIMM)

Table 168. Example Register Settings (Sheet 1 of 2)

Address	Row	Size of Row	Accumulative size	Register value
60h	Row 0 (DIMM 1, even)	512 MB	512 MB	08h
61h	Row 1 (DIMM 1, odd)	empty	512 MB	08h
62h	Row 2 (DIMM 2, even)	1 GB	1.5 GB	18h

Table 168. Example Register Settings (Sheet 2 of 2)

63h	Row 3 (DIMM 2, odd)	1 GB	2.5 GB	28hh
64h	Row 4 (Dimm 3, even)	1 GB	3.5 GB	38h
65h	Row 5 (Dimm 3, odd)	empty	3.5 GB	38h
66h	Row 6 (Dimm 4, even)	2 GB	5.5 GB	58h
67h	Row 7 (Dimm 4, odd)	2 GB	7.5 GB	78h

13.1.1.27 Offset 61h: DRB1 – DRAM Row 1 Boundary Register

Table 169. Offset 61h: DRB1 – DRAM Row 1 Boundary Register

Device: 0 ***Function:*** 0
Offset: *61h* ***Size:*** *8 bit*
Default Value: *00h*

Bits	Name	Description	Reset Value	Access
See Section 13.1.1.26, "Offset 60h: DRB0 – DRAM Row 0 Boundary Register" for details.				

13.1.1.28 Offset 62h: DRB2 – DRAM Row 2 Boundary Register

Table 170. Offset 62h: DRB2 – DRAM Row 2 Boundary Register

Device: 0 ***Function:*** 0
Offset: *62h* ***Size:*** *8 bit*
Default Value: *00h*

Bits	Name	Description	Reset Value	Access
See Section 13.1.1.26, "Offset 60h: DRB0 – DRAM Row 0 Boundary Register" for details.				

13.1.1.29 Offset 63h: DRB3 – DRAM Row 3 Boundary Register

Table 171. Offset 63h: DRB3 – DRAM Row 3 Boundary Register

Device: 0 ***Function:*** 0
Offset: *63h* ***Size:*** *8 bit*
Default Value: *00h*

Bits	Name	Description	Reset Value	Access
See Section 13.1.1.26, "Offset 60h: DRB0 – DRAM Row 0 Boundary Register" for details.				

13.1.1.30 Offset 64h: DRB4 – DRAM Row 4 Boundary Register

Table 172. Offset 64h: DRB4 – DRAM Row 4 Boundary Register

Device: 0			**Function:** 0	
Offset: 64h			**Size:** 8 bit	
Default Value: 00h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.1.1.26, "Offset 60h: DRB0 – DRAM Row 0 Boundary Register" for details.				

13.1.1.31 Offset 65h: DRB5 – DRAM Row 5 Boundary Register

Table 173. Offset 65h: DRB5 – DRAM Row 5 Boundary Register

Device: 0			**Function:** 0	
Offset: 65h			**Size:** 8 bit	
Default Value: 00h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.1.1.26, "Offset 60h: DRB0 – DRAM Row 0 Boundary Register" for details.				

13.1.1.32 Offset 66h: DRB6 – DRAM Row 6 Boundary Register

Table 174. Offset 66h: DRB6 – DRAM Row 6 Boundary Register

Device: 0			**Function:** 0	
Offset: 66h			**Size:** 8 bit	
Default Value: 00h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.1.1.26, "Offset 60h: DRB0 – DRAM Row 0 Boundary Register" for details.				

13.1.1.33 Offset 67h: DRB7 – DRAM Row 7 Boundary Register

The functionality of DRB7 is somewhat different from that of DRB[6:0]. If DRB(6:0) are non-zero and DRB7 is set to a value of 00h after memory configuration, the value is interpreted as 100h. This functionality avoids a 64 Mbyte "hole" at the top of memory. This behavior is unique to the DRB7 register.

Table 175. Offset 67h: DRB7 – DRAM Row 7 Boundary Register

Device: 0			**Function:** 0	
Offset: 67h			**Size:** 8 bit	
Default Value: 00h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.1.1.26, "Offset 60h: DRB0 – DRAM Row 0 Boundary Register" for details.				

13.1.1.34 Offset 70h: DRA0 – DRAM Row 0 Attribute Register

The DRAM Row Attribute register defines the DRAM technology and the DQ/DQS signal mapping to be used for each row of memory. Each nibble of information in the DRA registers describes the page size of a row. For this register, a row is defined by the chip select used by the DIMM, so that a dual rank DIMM has both an even and an odd entry. For single rank DIMMs, only the even side is used. See Table 177 on page 313.

Table 176. Offset 70h: DRA0 – DRAM Row 0 Attribute Register

Device: 0 ***Function:*** 0
Offset: *70h* ***Size:*** *8 bit*
Default Value: *00h*

Bits	Name	Description	Reset Value	Access
07:06	DWODD	**Device Width for odd-numbered row:** BIOS sets this bit according to the width of the DDR2 SDRAM devices populated in this row. This is used to determine the page size and the DQS to DQ signal mapping. 00 = Reserved 01 = x8 DDR2 (1 strobe pair per byte) 10 = x4 DDR2 (1 strobe pair per nibble) 11 = Reserved	00b	RW
05:04	DIMMTECH_ODD	**DIMM technology for odd-numbered row:** BIOS sets this bit according to the density of the DDR2 SDRAM devices populated in this row. This is used along with the device width to determine the page size. 00 = 2 Gbit DIMM 01 = Reserved 10 = 512 Mbit DIMM 11 = 1 Gbit DIMM	00b	RW
03:02	DWEVEN	**Device Width for even-numbered row:** BIOS sets this bit according to the width of the DDR2 SDRAM devices populated in this row. This is used to determine the page size and the DQS to DQ signal mapping. 00 = Reserved 01 = x8 DDR2 (1 strobe pair per byte) 10 = x4 DDR2 (1 strobe pair per nibble) 11 = Reserved.	00b	RW
01:00	DIMMTECH_EVEN	**DIMM technology for even-numbered row:** BIOS sets this bit according to the density of the DDR2 SDRAM devices populated in this row. This is used along with the Device Width to determine the page size and the DQS to DQ signal mapping. 00 = 2 Gbit DIMM 01 = Reserved 10 = 512 Mbit DIMM 11 = 1-Gbit DIMM	00b	RW

Table 177. DRA to DIMM Designation

	Even Row (or Single Rank)		Odd Row (Present if Dual Rank)	
	Row Number	Address of DRA	Row Number	Address of DRA
DIMM 1	Row 0	70h bits 03:00	Row 1	70h bits 07:04

Table 177. DRA to DIMM Designation

DIMM 2	Row 2	71h bits 03:00	Row 3	71h bits 07:04
DIMM 3	Row 4	72h bits 03:00	Row 5	72h bits 07:04
DIMM 4	Row 6	73h bits 03:00	Row 7	73h bits 07:04

13.1.1.35 Offset 71h: DRA1 – DRAM Row 1 Attribute Register

Table 178. Offset 71h: DRA1 – DRAM Row 1 Attribute Register

Device: 0 ***Function:*** 0
Offset: *71h* ***Size:*** *8 bit*
Default Value: *00h*

Bits	Name	Description	Reset Value	Access
See Section 13.1.1.34, "Offset 70h: DRA0 – DRAM Row 0 Attribute Register" for details. For DRAM rows 2 and 3.				

13.1.1.36 Offset 72h: DRA2 – DRAM Row 2 Attribute Register

Table 179. Offset 72h: DRA2 – DRAM Row 2 Attribute Register

Device: 0 ***Function:*** 0
Offset: *72h* ***Size:*** *8 bit*
Default Value: *00h*

Bits	Name	Description	Reset Value	Access
See Section 13.1.1.34, "Offset 70h: DRA0 – DRAM Row 0 Attribute Register" for details. For DRAM rows 4 and 5.				

13.1.1.37 Offset 73h: DRA3 – DRAM Row 3 Attribute Register

Table 180. Offset 73h: DRA3 – DRAM Row 3 Attribute Register

Device: 0 ***Function:*** 0
Offset: *73h* ***Size:*** *8 bit*
Default Value: *00h*

Bits	Name	Description	Reset Value	Access
See Section 13.1.1.34, "Offset 70h: DRA0 – DRAM Row 0 Attribute Register" for details. For DRAM rows 6 and 7.				

13.1.1.38 Offset 78 - 7Bh: DRT – DRAM Timing Register

This register controls the timing of the DRAM interface.

Table 181. Offset 78 - 7Bh: DRT – DRAM Timing Register (Sheet 1 of 6)

Device: 0	***Function:*** 0
Offset: *78 - 7Bh*	***Size:*** *32 bit*
Default Value: *859A_9604h*	

Bits	Name	Description	Reset Value	Access
31:30	PRGRPD	**Programmable Read Pointer Delay:** This bit determines the read pointer delay, which is based on both DIMM topology and technology. The round trip timing budget has been estimated to be about 10 ns. Since an encoding of "00" means less than one command clock, the encoding values in this table refer to additional delays beyond one command clock.	10b	RW
29:28	BTBWTRTA (tw2r)	**Back-To-Back Write-Read Turn Around:** This field determines the duration in CMDCLKs between write and read commands. It applies to any WR-RD pairs even to different rows. The purpose of these bits is to control the turn around time on DQ bus. Encoding — Command Clocks per Frequency 200 MHz 00 — 0 01 — 0 10 — 1 11 — 2	00b	
27:26	BTBRWTA (tr2w)	**Back-To-Back Read-Write Turn Around:** This field determines the minimum number of CMDCLK between Read-Write commands. It applies to RD-WR pairs to any destinations (in same or different rows). The purpose of this bit is to control the turnaround time on the DQ bus. Encoding — Command Clocks per Frequency 200 MHz 00 — 2 01 — 3 10 — 4 11 — 5	01b	RW

Table 181. Offset 78 - 7Bh: DRT – DRAM Timing Register (Sheet 2 of 6)

Device: 0 **Function:** 0
Offset: 78 - 7Bh **Size:** 32 bit
Default Value: 859A_9604h

Bits	Name	Description	Reset Value	Access
25:24	BTBRTA (tr2rdiff)	**Back To Back Read Turn Around: This field determines the minimum number of CMDCLK between two reads destined to different rows. The purpose of these bits is to control the turnaround time on the DQ bus.** Encoding: Command Clocks per Frequency 200 MHz 00: Reserved 01: 1 (5 ns) 10: 2 (10 ns) 11: 3 (15 ns)	01b	RW
23:22	Trfc	**Autorefresh cycle time:** The required tCK cycles between/after autorefresh cycles to any particular DIMM. NOTE: BIOS must always select largest memory technology when in configurations where different memory DIMMs are utilized. Encoding: Command Clocks per Frequency: memory technology 200 MHz: DDR2 00: 45 (225ns): 2Gb 01: 21 (105ns): 512Mb 10: 26 (130ns): 1Gb 11: Reserved: Reserved	10b	RW
21:20	Trrd	**Row Delay:** The required row delay period between two activate commands accessing the same CS of a DIMM in tCK cycles. Encoding: Command Clocks per Frequency 200 MHz 00: 1 01: 2 10: 3 11: 4	01b	RW

Table 181. Offset 78 - 7Bh: DRT – DRAM Timing Register (Sheet 3 of 6)

Device: 0 **Function:** 0
Offset: 78 - 7Bh **Size:** 32 bit
Default Value: 859A_9604h

Bits	Name	Description	Reset Value	Access
19:18	Trasmx	**Trasmax:** Indicates allowable number of clocks to allow sequential page hits prior to forcing a precharge to close the page. Encoding / Command Clocks per Frequency 200 MHz 00: 32 01: 64 10: 128 11: 512	10b	RW
17:16	Twr	**Write Recovery Delay:** The required write recovery delay before being able to issue a precharge to the same page accessing the same cs/bank of a DIMM in tCK cycles. Encoding / Command Clocks per Frequency 200 MHz 00: 1 01: 2 10: 3 11: 4	10b	RW
15:14	Trc	This bit controls the number of DRAM clocks to enforce as the RAS cycle time. Encoding / Command Clocks per Frequency 200 MHz 00: 11 01: 12 10: 13 11: 14	10b	RW

Table 181. Offset 78 - 7Bh: DRT – DRAM Timing Register (Sheet 4 of 6)

Device: 0 **Function:** 0
Offset: 78 - 7Bh **Size:** 32 bit
Default Value: 859A_9604h

Bits	Name	Description	Reset Value	Access
13:12	Tdal	**Write with Autoprecharge Recovery Delay:** The time required before being able to issue an activate to the same page accessing the same cs/bank of a DIMM Encoding / Command Clocks per Frequency 200 MHz 00: 6 01: 7 10: 8 11: 9	01b	RW
11:10	Trcd	**DRAM RAS# to CAS# delay:** This bits controls the number of clocks inserted between a row activate command and a read or write command to that row Encoding / Command Clocks per Frequency 200 MHz 00: 3 01: 4 10: 5 11: 6	01b	RW
09:08	Trp	**DRAM RAS# Precharge:** This bit controls the number of clocks that are inserted between a row precharge command and an activate command to the same row Encoding / Command Clocks per Frequency 200 MHz 00: Reserved 01: 3 10: 4 11: Reserved	10b	RW

Table 181. Offset 78 - 7Bh: DRT – DRAM Timing Register (Sheet 5 of 6)

Device: 0 **Function:** 0
Offset: 78 - 7Bh **Size:** 32 bit
Default Value: 859A_9604h

Bits	Name	Description	Reset Value	Access
07:06	BTBWTA	**Back-To-Back Write Turn Around:** This field determines the data bubble duration between write data bursts. It applies to WR-WR pairs to different ranks, and is only expected to be used in DDR2 mode with ODT enabled in the event that ODT selections must change between ranks. The purpose of this field is to control the data burst spacing on the DQ bus. Encoding: Command Clocks per Frequency 200 MHz 00: 0 (0 ns) 01: 1 (5 ns) 10: 2 (10 ns) 11: 3 (15 ns)	00b	RW
05	TACA	**Turn Around Cycle Add:** Setting this bit to a 1, adds an extra turn around cycle between a read to DIMM4 (furthest DIMM to IMCH) and a read to DIMM1 (nearest DIMM to IMCH).	0b	RW
04	CKEG	**CKE Guardband:** 0 = CKE is driven high for one CMDCLK prior to a new command 1 = CKE is driven high for two CMDCLKs prior to a new command	0b	RW
03:02	Tcl	**CAS# Latency:** The number of clocks between the rising edge used by DRAMS to sample the Read Command and the rising edge that is used by the DRAM to drive read data. Encoding: Command Clocks per Frequency 200MHz 00: 2 01: 3 10: 4 11: 5	01b	RW

Table 181. Offset 78 - 7Bh: DRT – DRAM Timing Register (Sheet 6 of 6)

Device: 0 **Function:** 0
Offset: 78 - 7Bh **Size:** 32 bit
Default Value: 859A_9604h

Bits	Name	Description	Reset Value	Access
01:00	CKE	**Idle Selection:** Specifies the number of CMDCLKs with no command activity for a row before deasserting CKE to '0'; putting the row into low power mode. Clocks of no command activity for a CS before dynamically driving CKE to '0'; forcing CS into power savings. Encoding: Command Clocks per Frequency 200 MHz 00: 32 01: 128 10: 512 11: 2 K	00b	RW

13.1.1.39 Offset 7C - 7Fh: DRC – DRAM Controller Mode Register

This register controls the mode of the DRAM Controller.The lower nibble of this register locks when the lower byte of the register is written for the first time and then it cannot be updated.

Table 182. Offset 7C - 7Fh: DRC – DRAM Controller Mode Register (Sheet 1 of 3)

Device: 0 **Function:** 0
Offset: 7C - 7Fh **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:30	Reserved	Reserved	00b	
29	IC	0 = **Initialization Complete:** This bit is used for communication of software state between the memory controller and the BIOS. DRAM interface has not been initialized. 1 = DRAM interface has been initialized.	0b	RW
28	Reserved	**Reserved**	0b	
27	DEDRTRY	**DED Retry Enable:** 0 = Disabled. No retries occur on double-bit errors. 1 = Enable a single retry of read accesses on detection of a double-bit error.	0b	RW
26	OVREN	**Overlap enable:** 0 = Inhibits overlap scheduling of row/columns tenures. 1 = Allows overlapped scheduling of activates prior to completing the outstanding column command.	0b	RW

Table 182. Offset 7C - 7Fh: DRC – DRAM Controller Mode Register (Sheet 2 of 3)

Device: 0 **Function:** 0
Offset: 7C - 7Fh **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
25:24	APMW	**Auto-precharge mode for writes:** 00 Intelligent 01 Always Auto-precharge 10 Never Auto-precharge 11 Reserved	00b	RW
23:22	APMR	**Auto-precharge mode for reads:** 00 Intelligent 01 Always Auto-precharge 10 Never Auto-precharge 11 Reserved	00b	RW
21:20	DDIM	**DRAM Data Integrity Mode:** 00 reserved 01 72-bit ECC 10 Reserved 11 Reserved	00b	RW
19:11	Reserved	Reserved	00b	
10:08	RMS	**Refresh Mode Select:** This field determines whether refresh is enabled and, if so, at what rate refreshes are executed. 000 Refresh disabled. 001 Refresh enabled. Refresh interval 15.6 µs 010 Refresh enabled. Refresh interval 7.8 µs 011 Refresh enabled. Refresh interval 64 µs 100 Refresh enabled. Refresh interval 3.9 µs 101) Reserved 110 Reserved 111 Refresh enabled. Refresh interval 64 clocks Others Reserved)	000b	RW
07	Reserved	Reserved	0b	
06	CMDDISA	**CMD Disable [Sticky]:** 0 = CMD enabled 1 = CMD disabled	0b	RW
05	DRAMODT	**DRAM ODT Disable [STICKY]:** 0 = Enables the use of ODT when running 1 = Disables the use of ODT when running	0b	RW
04	CKEPNM	**CKE pin mode [STICKY]:** DDR2 Clock Enables have two operating modes: 0 = Disable: CKEs are disabled during reset. 1 = . Enable: BIOS will set this bit to a 1 for normal operating mode.	0b	RW

Table 182. Offset 7C - 7Fh: DRC – DRAM Controller Mode Register (Sheet 3 of 3)

Device: 0	***Function:*** 0
Offset: *7C - 7Fh*	***Size:*** *32 bit*
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
03:02	FSBFREQSEL	**Front Side Bus Frequency Select in MHz [STICKY].** Due to the use of these bits to define internal clocking, they must be sticky through reset. Default 10b, for 167 MHz primary production target frequency.The PLL only supports one update of ratio (the lower nibble of this register). Failure to properly program these bits can result in the inability to access configuration space. Bit Field Encoding 00 100 MHz 01 133 MHz 10 167 MHz 11 Reserved. The lower nibble of this register locks only when the lower byte of the register is written for the first time. Then it cannot be further updated. **Note**: BIOS must ensure the correct value is programmed for these bits or indeterminate results will occur.	10b	RWO
01:00	DT	**DRAM Type [STICKY]:** 00 Reserved 01 Reserved 10 DDR2-400 11 Reserved BIOS will write 10 to these bits	00b	RWO

13.1.1.40 Offset 80 - 81h: DRM – DRAM Mapping Register

This register is used for mapping CS logical to CS physical. See the *RS* - Intel® 3100 Chipset *BIOS Specification* for more details for this mapping.

The one-hot encodings are the normal modes. The instrumentation mode is only used when a DIMM is not present in a slot.

Table 183. Offset 80 - 81h: DRM – DRAM Mapping Register

Device: 0 **Function:** 0
Offset: 80 - 81h **Size:** 16 bit
Default Value: 8421h

Bits	Name	Description	Reset Value	Access
15:12	DIMM1	**Logical CS to physical CS[07:06] mapping [STICKY]:** 0001 Logical CS(01:00) maps to physical CS(07:06) 0010 Logical CS(03:02) maps to physical CS(07:06) 0100 Logical CS(05:04) maps to physical CS(07:06) 1000 Logical CS(07:06) maps to physical CS(07:06) 1111 Global CS (Instrumentation mode) maps to physical CS(07:06)	8h	RW
11:08	DIMM2	**Logical CS to physical CS[05:04] mapping [STICKY]:** 0001 Logical CS(01:00) maps to physical CS(05:04) 0010 Logical CS(03:02) maps to physical CS(05:04) 0100 Logical CS(05:04) maps to physical CS(05:04) 1000 Logical CS(07:06) maps to physical CS(05:04) 1111 Global CS (Instrumentation mode) maps to physical CS(05:04)	4h	RW
07:04	DIMM3	**Logical CS to physical CS[03:02] mapping [STICKY]:** 0001 Logical CS(01:00) maps to physical CS(03:02) 0010 Logical CS(03:02) maps to physical CS(03:02) 0100 Logical CS(05:04) maps to physical CS(03:02) 1000 Logical CS(07:06) maps to physical CS(03:02) 1111 Global CS (Instrumentation mode) maps to physical CS(03:02)	2h	RW
03:00	DIMM4	**Logical CS to physical CS[01:00] mapping [STICKY]:** 0001 Logical CS(01:00) maps to physical CS(01:00) 0010 Logical CS(03:02) maps to physical CS(01:00) 0100 Logical CS(05:04) maps to physical CS(01:00) 1000 Logical CS(07:06) maps to physical CS(01:00) 1111 Global CS (Instrumentation mode) maps to physical CS(01:00)	1h	RW

13.1.1.41 Offset 82h: DRORC – Opportunistic Refresh Control Register

The IMCH contains a 4b refresh counter that allows the counting of up to 16 refreshes. Using the counter, refresh requests can be queued when the DRAM interface is busy performing cycles. Ideally, refreshes are performed when the DRAM interface is idle. This opportunistic refresh scheme utilizes two watermarks, which the following register is used to control.

This register is used for diagnostic testing of ECC from the DRAM.

Table 184. Offset 82h: DRORC – Opportunistic Refresh Control Register

Device: 0	**Function:** 0
Offset: 82h	**Size:** 8 bit
Default Value: 71h	

Bits	Name	Description	Reset Value	Access
07:04	HIGHW	**High Watermark:** When the refresh-counter reaches or exceeds the value in the high watermark field, the DRAM controller performs a refresh in the highest priority mode. In such a case, refresh is processed as soon as the currently pending DRAM cycle is completed. Once a high priority refresh is internally launched (through the command queue), the DRAM controller may schedule an additional refresh immediately if the refresh counter high watermark condition remains "true". Current DDR2 components require DLL refresh every nine refresh periods. As a result, this register must be set at seven or lower. Bit field encoding: 0000 Illegal value 0001 One Refresh is the watermark ..... 1111 15 Refreshes is the watermark	0111b	RW
03:00	LOWW	**Low Watermark:** When the refresh-counter reaches or exceeds the value in the low watermark field, the DRAM controller performs a refresh if there is no other request pending to DRAM. It means that low watermark refresh is performed as the lowest priority request, opportunistically. Once a low priority refresh is internally launched (through the command queue), the DRAM controller does not schedule an additional low priority refresh until the already launched refresh operation is completed (low watermark refresh is blocked when the command queue contains a low watermark refresh request). Once low watermark refresh counter is reached or exceeded, the DRAM controller opportunistically performs low priority refreshes until the refresh counter is down to 0. Bit field encoding: 0000 Illegal value 0001 One Refresh is the watermark ..... 1111 15 Refreshes is the watermark	0001b	RW

13.1.1.42 Offset 84 - 87h: ECCDIAG – ECC Detection/Correction Diagnostic Register

This register is used for diagnostic testing of ECC to the DRAM. This feature is presented for validation purposes only. Functionality is not guaranteed and may not be supported.

Table 185. Offset 84 - 87h: ECCDIAG – ECC Detection/Correction Diagnostic Register

Device: 0 **Function:** 0
Offset: 84 - 87h **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:21	Reserved	Reserved	000h	
20	FECCDT	**Flip ECC on all data transfers:** Flip the designated ECC bits (bits 15:00) on all data transfers to DRAM. If a cacheline is in progress when this register is written, wait until the start of the next cacheline to flip parity bits.	0b	RW
19	FECCNDT	**Flip ECC on only the next data transfer:** Flip the designated ECC bits (bits 15:00) in only the next data transfer to DRAM. If a cacheline is in progress when this register is written, wait until the start of the next cacheline to flip parity bits to ensure all bits flipped are within the same cacheline. Hardware clears this bit when the injection has been performed.	0b	RWS
18	MEMPEN	**Memory Poison Enable:** Allows for propagation of data errors not initiated by this feature to DRAM. Error injection via bits 19 or 20 is possible regardless of this bit setting. 0 = Error poisoning is disabled, data errors are not propagated, meaning that only good ECC is generated when in 72-bit tECC mode. 1 = Error poisoning enabled when in either 72-bit ECC mode.	0b	RW
17:16	DPRSL	**Data pair selector:** This two-bit field selects which pair of quad-words (QW) in a cache line the inversion vector is applied against. Regardless of what operational mode the memory subsystem is in, this field always applies to the same QW pair. QW0 corresponds to data bits 63:00, QW1 to [127:64] … and QW7 corresponds to [511:448] 00 QW0 and QW1 01 QW2 and QW3 10 QW4 and QW5 11 QW6 and QW7	00b	RW
15:00	ECCBIN	**ECC bit invert vector:** This vector operates individually for every ECC bit in the selected High or Low ECC block, during writes to DRAM. For all k between 0 and 15, when bit (k) set to 1, the value of the k ECC bit (which corresponds with the k data byte lane) is inverted. Otherwise, the value the k ECC bit is not affected. In other words, bits 15:08 are applied to the ECC vector of the high Qword in the selected pair, and bits 07:00 are applied to the ECC vector of the low Qword in the selected pair. For Example: Data Pair Selector bits 17:16 = 00 ECC bit invert vector bits 15:08 are applied to the ECC vector for QW1 ECC bit invert vector bits 07:00 are applied to the ECC vector for QW0	0000h	RW

13.1.1.43 Offset 88 - 8Bh: SDRC – Secondary DRAM Controller Mode Register

Table 186. Offset 88 - 8Bh: SDRC – Secondary DRAM Controller Mode Register (Sheet 1 of 2)

Device: 0 **Function:** 0
Offset: 88 - 8Bh **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:30	Reserved	Reserved	00b	
29:28	ODTENA	**On Die Termination Enable:** These bits enable the IMCH on-die termination for the DDR2 data signals. Encoding / DDR2 IMCH ODT (Rodt = 285) 00: Off / Off 01: Rodt/2 / ~150 10: Rodt/2 / ~150 11: Rodt/4 / ~75	00b	RW
27:26	Reserved	Reserved	00b	
25	CQDO	**Command Queue Depth One:** 0 = This bit causes the queue to function with a depth of one. 1 = The command queue depth is four cycles (normal operating mode).	0b	RW
24	Legsel	**Testmode:** 0 = The number of driver slices enabled is equal to legsel value plus one (normal operating mode). 1 = Legsel output determines which one of seven driver slices would be enabled. ***Note:*** Depending on the value of DRAMISCTL[31:27], you may either enable compensation and get the legsel value generated by the compensation block logic or override the value generated by that block with the one provided by the configuration register.	0b	RW
23:09	Reserved	Reserved	0000h	
08	DQSHALFGAIN	**Select for the DQS differential amplifier gain**. Expected to be set to 1 to cut the gain roughly in half for differential strobe mode associated with DDR2.	0b	RW
07	DIFFDQSEN	**Differential DQS enable**. Must be set to 1 for DDR2 when programmed to use differential data strobes.	0b	RW

Table 186. Offset 88 - 8Bh: SDRC – Secondary DRAM Controller Mode Register (Sheet 2 of 2)

Device: 0	**Function:** 0
Offset: 88 - 8Bh	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
06:04	SFM	**The Spatial Flush Mask:** This an 8-bit field that corresponds to the address fields specified by the DDRCSR register. The bit field selects which bits is used as the uppermost enabled mask bit: 00000001 0000001- 000001- - 00001- - - 0001- - - - 001- - - - - 01- - - - - - 1- - - - - - - The width (bits 03:02) is used to define which of the '-' bits are also asserted.	000b	RW
03:02	SFWSEL	**Spatial flush width select:** This width is used to determine the additional number of asserted bits to the right of the selected MSB bit. Together these construct a mask that has a contiguous set of 1 value of variable width and position within the 8-bit flush mask. Refer to Chapter 12.0, "Supported DRAM Technology" for details.	00b	RW
01:00	SFMD	**Spatial Flush Mode:** This register is used to control the Write Cache flush mode used in the IMCH. The write cache selects write entries to be flushed to DRAM based on two rules, temporal and spatial. Initial writes or when writes don't meet spatial rules, are selected to be flushed based on LRU scheme. Otherwise writes to be flushed are selected based on spatial rules. The DDR2 controller supports two spatial rules: Page Hit: write cache searches for page hit cycles to be next flushed to DRAM. Row Miss: write cache searches for row miss cycles to be next selected in the flush stream. The setting of this bit field: 1x Spatial Flush Mode is disabled. Writes are flushed based on LRU scheme. 01 Row Miss Spatial Flush Mode is enabled (combination of LRU and row miss scheme). 00 Page Hit Spatial Flush Mode is enabled (combination of LRU and page hit scheme).	00b	RW

13.1.1.44 Offset 8Ch: CKDIS – CK/CK# Clock Disable Register

Table 187. Offset 8Ch: CKDIS – CK/CK# Clock Disable Register

Device: 0	**Function:** 0
Offset: 8Ch	**Size:** 8 bit
Default Value: FFh	

Bits	Name	Description	Reset Value	Access
07:00	CLKDIS	**CK/CK# Disable [STICKY]:** Each bit corresponds to a CMDCLK/CMDCLK# pair of pins on one of the DIMMs, 0 = Enable CMDCLK signals to the corresponding DIMM (or DIMM pair) 1 = Disable CMDCLK signals to the corresponding DIMM (or DIMM pair). When disabled, the corresponding CMDCLK/ CMDCLK# pair are tri-stated. The list below shows how the clock pins are designated. **Bit DIMM** 7 Reserved 6 Reserved 5 Reserved 4 Reserved 3 DDRA_CMDCLK3 / DDRA_CMDCLK3# 2 DDRA_CMDCLK2 / DDRA_CMDCLK2# 1 DDRA_CMDCLK1 / DDRA_CMDCLK1# 0 DDRA_CMDCLK0 / DDRA_CMDCLK0#	FFh	RW

13.1.1.45 Offset 8Dh: CKEDIS – CKE/CKE# Clock Disable Register

Table 188. Offset 8Dh: CKEDIS – CKE/CKE# Clock Disable Register

Device: 0	**Function:** 0
Offset: 8Dh	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07:00	CKEDIS	**CKE disable [STICKY]:** Each bit corresponds to a CKE pin as defined in the table below. 0 = Enable CKE signal (Default) 1 = Disable CKE signal Bit Device Pin 7 Reserved 6 CKE3 5 Reserved 4 CKE2 3 Reserved 2 CKE1 1 Reserved 0 CKE0	00	RW

13.1.1.46 Offset 90 - 93h: SPARECTL – SPARE Control Register

Table 189. Offset 90 - 93h: SPARECTL – SPARE Control Register

Device: 0	***Function:*** 0
Offset: *90 - 93h*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
31:24	DEDEPV	**DED error prescale value:** Prescale value ranges from 0-255.	00h	RW
23:16	SECEPV	**SEC error prescale value:** Prescale value ranges from 0-255.	00h	RW
15:12	DEDEPU	**DED error prescale unit:** 0000 Never 0001 1 µs 0010 1 ms 0011 1 s 0100 1 minute 0101 1 hour 0110 1 day 0111 1 week 1XXX Never	0h	RW
11:08	SECEPU	**SEC error prescale unit:** 0000 Never 0001 1 µs 0010 1 ms 0011 1 s 0100 1 minute 0101 1 hour 0110 1 day 0111 1 week 1XXX Never	0h	RW
07:00	Reserved	Reserved	00h	

13.1.1.47 Offset 94 - 97h: DRAMISCTL – DRAM Miscellaneous Control Register

This register contains specialized bits to control operation of custom DDR2 interface circuitry.

Table 190. Offset 94 - 97h: DRAMISCTL – DRAM Miscellaneous Control Register

Device: 0 **Function:** 0
Offset: 94 - 97h **Size:** 32 bit
Default Value: B030_0000h

Bits	Name	Description	Reset Value	Access
31:27	DRVOVR	This field controls the behavior of the DDR2 driver impedance. 0nnnn = enable compensation Target impedance as selected by "nnnn" with specified external impedance control resistor. This resistor is roughly equally to the pull-up and pull-down resistors in the driver. **nnnn** **DDR2 w/ 285 Ohm** 1010 28.5 ohms 1001 23.8 ohms 1000 20.4 ohms 0111 17.8 ohms 0110 15.8 ohms 0101 NA 0100 NA 0011 NA 0010 NA	16h	RW
26:25	Reserved	Reserved	00b	
24	VOXSTART	Enables voltage output crossing control loop. This bit must always be set by BIOS for normal operation, but only after clocks have been enabled.	0b	RW
23	Reserved	Reserved	0b	
22	OCDLOADEN	Calibration buffer load placed on incoming signals to perform calibration. Required for OCD calibration when DDR2 ODT mode is in use.	0b	RW
21	Reserved	Reserved	1b	
20	CHASLVBYP	Bypasses DLL for strobe calibration. Bypass enabled at reset.	1b	RW
19:17	Reserved	Reserved	000b	
16:14	CHASLVLEN	Minimum delay setting for all accesses for all nibbles. Measured in 1/8th DDR2 clock increments (0-7).	000b	RW
13:08	Reserved	Reserved	00b	
07:04	Reserved	Reserved	0h	
03:00	CHAVREF	Controls the VREF selection/generation mechanism. Vref calibration is a required BIOS function. 0 External Vref (default) 1 Internal vref -250 mV 2 Internal vref -200 mV 3 Internal vref -150 mV 4 Internal vref -100 mV 5 Internal vref -50 mV 6 Internal vref -10 mV 7 Internal precision VREF 8 Internal vref +10 mV 9 Internal vref +50 mV A Internal vref +100 mV B Internal vref +150 mV C Internal vref +200 mV D Internal vref +250 mV E Pull-down Calibration mode F Pull-up Calibration mode	0h	RW

13.1.1.48 Offset 9A - 9Bh: DDRCSR – DDR Channel Configuration Control/Status Register

Table 191. Offset 9A - 9Bh: DDRCSR – DDR Channel Configuration Control/Status Register

Device: 0	***Function:*** 0
Offset: *9A - 9Bh*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15	TRANEN	**Transition Enable:** 0 = No change. This bit is cleared by the IMCH when the Channel Configuration state change has occurred. 1 = Set and enable Channel Configuration transitions,	0b	RWS
14:12	FSMISTQ	**Idle State Transition Qualifier:** 000 Idle 001 Transition to Memory Channel State All other transitions are reserved.	000b	RW
11:04	Reserved	Reserved	00b	
03:00	CCFSMCS	**Channel Configuration Current State:** This field can be read to verify that the correct state transition occurred. State Description: 0000 Idle 0100 Memory Channel (Normal) state All other sates are reserved.	0000b	RO

13.1.1.49 Offset 9Ch: DEVPRES – Device Present Register

This register can only be written to at boot time when there is no traffic to or from the PCI Express ports. The Device Present bits can be used to enable/disable devices within the IMCH and make their PCI configuration space respectively visible/invisible to software. Once software or BIOS has written this register for the first time after power-up, the register value locks, and cannot be further updated.

In other words, once software has disabled devices, they can only be re-enabled via a reset. The IMCH does not support turning off a device, and then turning it back on. (The reverse is also true: once software has enabled devices, they can only be re-disabled via a reset.)

When a bit is 0, the configuration space associated with that device is hidden, returning all 1's for all configuration register reads just as if the cycle terminated with a master abort on PCI. The I/O buffers associated with that device are disabled and tri-stated. Compensation is disabled. When a bit is 1, the configuration space associated with that device is accessible. The I/O buffers are enabled. Compensation is enabled.

Note: BIOS must write a bit to a 0 to disable a device, or a 1 to enable a device. This register can only be written once. After the first write, the register is locked.

Note: BIOS should write this register as part of its power on configuration sequence

Table 192. Offset 9Ch: DEVPRES – Device Present Register

Device: 0	**Function:** 0
Offset: 9Ch	**Size:** 8 bit
Default Value: 03h	

Bits	Name	Description	Reset Value	Access
07	Reserved	Reserved	0b	
06	Reserved	Reserved	0b	
05	Reserved	Reserved	0b	
04	Reserved	Reserved	0b	
03	Device 3 Present	0 = PCI Express* port A1 (x4) is disabled. In this state, port A (Device 2) can operate with a maximum x8 link width. 1 = PCI Express port A1 is enabled. In this state, port A can operate with a maximum x4 link width.	0b	RWO
02	Device 2 Present	0 = PCI Express port A (x8) or port A0 (x4) is disabled. 1 = PCI Express port A (x8) or port A0 (x4) is enabled.	0b	RWO
01	Device 1 Present	0 = EDMA Controller is disabled. 1 = EDMA Controller is enabled. This bit is forced to 0 when the corresponding bit is set.	1b	RWO
00	Reserved	Reserved	1b	

13.1.1.50 Offset 9Dh: EXSMRC – Extended System Management RAM Control Register

The Extended SMRAM register controls the configuration of Extended SMRAM space. The Extended SMRAM (E_SMRAM) memory provides a write-back cacheable SMRAM memory space that is above 1 MByte.

Table 193. Offset 9Dh: EXSMRC – Extended System Management RAM Control Register (Sheet 1 of 3)

Device: 0	**Function:** 0
Offset: 9Dh	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07	H_SMRAME	**Enable High SMRAM:** Controls the SMM memory space location (above 1 MByte or below 1 MByte) 0 = High SMRAM memory space is disabled. 1 = If G_SMRAME is 1, the high SMRAM memory space is enabled. SMRAM accesses within the range 0FEDA_0000h to 0FEDB_FFFFh are remapped to DRAM addresses within the range 000A0000h to 000BFFFFh. Once D_LCK (See Table 194) has been set, this bit becomes Read-Only.	0b	RWL

Table 193. Offset 9Dh: EXSMRC – Extended System Management RAM Control Register (Sheet 2 of 3)

Device: 0	**Function:** 0
Offset: 9Dh	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
06	MDAP	**MDA Present:** This bit works with the VGA Enable bits in the BCTRL registers of Devices 2–3 to control the routing of processor initiated transactions targeting MDA compatible I/O and memory address ranges. This bit should not be set if none of the VGA Enable bits are set. If none of the VGA enable bits are set, then accesses to IO address range x3BCh-x3BFh are forwarded to NSI. If the VGA enable bit is not set then accesses to IO address range x3BCh-x3BFh are treated just like any other IO accesses. For example, the cycles are forwarded to PEA[0:1] if the address is within the corresponding IOBASE and IOLIMIT and ISA enable bit is not set, otherwise they are forwarded to NSI. ***Note:*** Since the logic performs the address decoding on a DW boundary, the DW that includes the address 3BF also includes addresses 3BC, 3BD, and 3BE, and accesses to any of these byte addresses are handled as MDA references. MDA resources are defined as the following: Memory: 0B0000h - 0B7FFFh I/O: 3B4h, 3B5h, 3B8h, 3B9h, 3BAh, 3BFh, (Including ISA address aliases, A[15:10] are not used in decode) ***Note:*** The VGA region includes I/O space ranges 3B0-3BBh, and 3C0-3DFh, so there is an overlap between these two I/O regions. Any I/O reference that includes the I/O locations listed above, or their aliases, are forwarded to NSI even if the reference includes I/O locations not listed above. The following table shows the behavior for all combinations of MDA and VGA: **VGA MDA Behavior** 0 0 All References to MDA and VGA go to NSI 0 1 Illegal Combination (DO NOT USE) 1 0 All References to VGA go to device with VGA enable set. MDA- only references (I/O address 3BF and aliases) go to NSI. 1 1 VGA-only references go to the PCI Express port which has its VGA Enable bit set. MDA references go to the NSI.	0b	RW
05	APICDIS	**APIC Memory Range Disable:** 0 = The IMCH send cycles between 0_FEC0_0000 and 0_FEC7_FFFF to NSI, accesses between 0_FEC8_0000 and 0_FEC8_0FFF are sent to PEA0, between 0_FEC8_1000 and 0_FEC8_1FFF are sent to PEA1. 1 = The IMCH forwards all accesses to the IOAPIC regions through NSI.	0b	RW
04	Reserved	**Reserved**	0b	
03	G_SMRAME	**Global SMRAM Enable:** 0 = The Compatible SMRAM functions are disabled. 1 = The Compatible SMRAM functions are enabled, providing 128 Kbyte of DRAM accessible at the A0000h address while in SMM (ADS# with SMM decode). To enable Extended SMRAM function this bit has be set to 1. Refer to Section 5.5 for more details. Once D_LCK (See Table 194) is set, this bit becomes read-only.	0b	RWL

Table 193. Offset 9Dh: EXSMRC – Extended System Management RAM Control Register (Sheet 3 of 3)

Device: 0	**Function:** 0
Offset: 9Dh	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
02:01	TSEG_SZ	**TSEG Size:** Selects the size of the TSEG memory block if enabled. Memory from the top of DRAM space (TOLM - TSEG_SZ) to TOLM is partitioned away so that it may only be accessed by the processor interface and only then when the SMM bit is set in the request packet. Non-SMM accesses to this memory region are specially terminated when the TSEG memory block is enabled. Note that once D_LCK (See Table 194) is set, these bits become Read-Only. 0 0 (TOLM – 128 k) to TOLM 0 1 (TOLM – 256 k) to TOLM 1 0 (TOLM – 512 k) to TOLM 1 1 (TOLM – 1 M) to TOLM	00b	RWL
00	T_EN	**TSEG Enable:** Enabling of SMRAM memory for Extended SMRAM space only. 0 = SMRAM memory for Extended SMRAM space disabled. 1 = And G_SMRAME =1 and T_EN = 1, the TSEG is enabled to appear in the appropriate physical address space. Once D_LCK (See Table 194) is set, this bit becomes Read-Only.	0b	RWL

13.1.1.51 Offset 9Eh: SMRAM – System Management RAM Control Register

The SMRAMC register controls how accesses to Compatible and Extended SMRAM spaces are treated. The Open, Close, and Lock bits function only when G_SMRAME bit is set to a 1. Also, the OPEN bit must be reset before the LOCK bit is set.

Table 194. Offset 9Eh: SMRAM – System Management RAM Control Register (Sheet 1 of 2)

Device: 0	**Function:** 0
Offset: 9Eh	**Size:** 8 bit
Default Value: 02h	

Bits	Name	Description	Reset Value	Access
07	Reserved	Reserved	0b	
06	D_OPEN	**SMM Space Open:** 0 = The SMM space DRAM is not visible 1 = And D_LCK=0, the SMM space DRAM is made visible even when SMM decode is not active. This is intended to help BIOS initialize SMM space. Software must ensure that D_OPEN=1 and D_CLS=1 are not set at the same time. This bit becomes RO when D_LCK is set to 1.	0b	RWL

Table 194. Offset 9Eh: SMRAM – System Management RAM Control Register (Sheet 2 of 2)

Device: 0	**Function:** 0
Offset: 9Eh	**Size:** 8 bit
Default Value: 02h	

Bits	Name	Description	Reset Value	Access
05	D_CLS	**SMM Space Closed:** 0 = SMM space DRAM is accessible to data references 1 = SMM space DRAM is not accessible to data references, even if SMM decode is active. Code references may still access SMM space DRAM. This allows SMM software to reference through SMM space to update the display even when SMM is mapped over the VGA range. Software must ensure that D_OPEN=1 and D_CLS=1 are not set at the same time. ***Note:*** The D_CLS bit only applies to Compatible SMM space.	0b	RW
04	D_LCK	**SMM Space Locked:** 0 = SMM space unlocked 1 = And then D_OPEN is reset to 0 and D_LCK, D_OPEN, H_SMRAME, TSEG_SZ and T_EN become Read-Only. D_LCK can be set to 1 via a normal configuration space write but can only be cleared by a Full Reset. The combination of D_LCK and D_OPEN provide convenience with security. The BIOS can use the D_OPEN function to initialize SMM space and then use D_LCK to lock SMM space in the future so that no application software (or BIOS itself) can violate the integrity of SMM space, even if the program has knowledge of the D_OPEN function.	0b	RWS
03	Reserved	Reserved.	0b	
02:00	C_BASE_SEG	**Compatible SMM Space Base Segment:** This field indicates the location of SMM space. SMM DRAM is not remapped. It is simply made visible if the conditions are right to access SMM space, otherwise the access is treated as a VGA access. Since the IMCH supports only the SMM space between A0000 and BFFFF, this field is hardwired to 010.	010b	RO

13.1.1.52 Offset 9Fh: EXSMRAMC – Expansion System Management RAM Control Register

The Extended SMRAM register controls the configuration of Extended SMRAM space. The Extended SMRAM (E_SMRAM) memory provides a write-back cacheable SMRAM memory space that is above 1 MByte.

Table 195. Offset 9Fh: EXSMRAMC – Expansion System Management RAM Control Register

Device: 0	***Function:*** 0
Offset: *9Fh*	***Size:*** *8 bit*
Default Value: *07h*	

Bits	Name	Description	Reset Value	Access
07	E_SMERR	**Invalid SMRAM Access:** 0 = Processor has not accessed the defined memory ranges in Extended SMRAM. 1 = This bit is set when processor has accessed the defined memory ranges in Extended SMRAM (High Memory and T-segment) while not in SMM space and with the D-OPEN bit = 0. It is software's responsibility to clear this bit. This bit is cleared by software writing a 1 to the bit location.	0b	RWC
06:03	Reserved	Reserved	0h	
02	SM_CACHE	**SMRAM Cacheable: This bit is forced to 1 by** IMCH**.**	1b	RO
01	SM_L1	**L1 Cache Enable for SMRAM:** This bit is forced to 1 by IMCH.	1b	RO
00	SM_L2	**L2 Cache Enable for SMRAM:** This bit is forced to 1 by IMCH.	1b	RO

13.1.1.53 Offset A0 - A3h: CLKGRFM0 – Clock Gearing Ratio FSB to Memory 0 Register

This register consists of eight nibbles of mux select for the proper selection of gearing behavior. This is the first of two registers to control the behavior for the FSB to Memory direction of data flow. This register is also referred to as the primary register of the pair. These bits are sticky through reset.

Table 196. Offset A0 - A3h: CLKGRFM0 – Clock Gearing Ratio FSB to Memory 0 Register

Device: 0	***Function:*** 0
Offset: *A0 - A3h*	***Size:*** *32 bit*
Default Value: 0015_4320h	

Bits	Name	Description	Reset Value	Access
31:00	**CLKGRFM)**	**Clock Gearing mux selects [STICKY]:** Eight nibbles of mux select for FSB/core to memory/DDR2 geared clock boundary crossing phase enables. FSB: Memory Frequency Ratio Register Value 100:200 1:2 0000_0010 133:200 2:3 0000_0120 167:200 5:6 0015_4320 Note: BIOS must ensure the correct value is programmed for these bits or indeterminate results will occur.	0015_4320h	RW

13.1.1.54 Offset A4 - A7h: CLKGRFM1 – Clock Gearing Ratio FSB to Memory 1 Register

This register consists of eight nibbles of mux select for the proper selection of gearing behavior. This is the second of two registers to control the behavior for the FSB to Memory direction of data flow. This register is also referred to as the secondary register of the pair. These bits are sticky through reset.

Table 197. Offset A4 - A7h: CLKGRFM1 – Clock Gearing Ratio FSB to Memory 1 Register

Device: 0 ***Function:*** 0
Offset: *A4 - A7h* ***Size:*** *32 bit*
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:00	CLKGRFM1	**Clock Gearing mux selects [STICKY]:** Eight nibbles of mux select for FSB/core to memory/DDR2 geared clock boundary crossing phase enables. FSB: Memory Frequency Ratio Register Value 100:200 1:2 0000_0000 133:200 2:3 0000_0000 167:200 5:6 0000_0000 Note: BIOS must ensure the correct value is programmed for these bits or indeterminate results will occur.	0000_0000h	RW

13.1.1.55 Offset A8 - ABh: CLKGRMF0 – Clock Gearing Ratio Memory to FSB 0 Register

This register consists of eight nibbles of mux select for the proper selection of gearing behavior. This is the first of two registers to control the behavior for the Memory to FSB direction of data flow. This register is also referred to as the primary register of the pair. These bits are sticky through reset.

Table 198. Offset A8 - ABh: CLKGRMF0 – Clock Gearing Ratio Memory to FSB 0 Register

Device: 0 ***Function:*** 0
Offset: *A8 - ABh* ***Size:*** *32 bit*
Default Value: 0006_5432h

Bits	Name	Description	Reset Value	Access
31:00	CLKGRMF0	**Clock Gearing mux selects [STICKY]:** Eight nibbles of mux select for memory/DDR2 to FSB/core geared clock boundary crossing phase enables. FSB: Memory Frequency Ratio Register Value 100:200 1:2 0000_0002 133:200 2:3 0000_0032 167:200 5:6 0006_5432	0006_5432h	RW

13.1.1.56 Offset AC - AFh: CLKGRMF1 – Clock Gearing Ratio Memory to FSB 1 Register

This register consists of eight nibbles of mux select for the proper selection of gearing behavior. This is the second of two registers to control the behavior for the Memory to FSB direction of data flow. This register is also referred to as the secondary register of the pair. These bits are sticky through reset.

Table 199. Offset AC - AFh: CLKGRMF1 – Clock Gearing Ratio Memory to FSB 1 Register

Device: 0 ***Function:*** 0
Offset: *AC - AFh* ***Size:*** *32 bit*
Default Value: 0001_0000h

Bits	Name	Description	Reset Value	Access
31:00	CLKGRMF1	**Clock Gearing mux selects [STICKY]:** Eight nibbles of mux select for memory/DDR2 to FSB/core geared clock boundary crossing phase enables. FSB: Memory Frequency Ratio Register Value 100:200 1:2 0000_0001 133:200 2:3 0000_0010 167:200 5:6 0001_0000 Note: BIOS must ensure the correct value is programmed for these bits or indeterminate results will occur.	0001_0000h	RW

13.1.1.57 Offset B0 - B3h: DDR2ODTC – DDR2 ODT Control Register

This register consists of four bytes of ODT enable vectors, one per DDR2 rank, each of which contains a nibble for read accesses and a nibble for write accesses. The encodings for each nibble are a reflection of the four ODT outputs driven on the odd CS lines of the channel during an access to the corresponding rank. This register definition is dependent on the setting of the DRM register, because it is both a logical ODT vector and a logical CS that is selected. Refer to the lower nibble bit field description for clarification.

Table 200. Offset B0 - B3h: DDR2ODTC – DDR2 ODT Control Register (Sheet 1 of 2)

Device: 0 ***Function:*** 0
Offset: *B0 - B3h* ***Size:*** *32 bit*
Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:28	R4ODTWR	**R4ODTWR.** Logical ODT vector for write access to logical rank 4 (CS 6). Refer to field 03:00 description for encodings.	0h	RW
27:24	R4ODTRD	**R4ODTRD**. Logical ODT vector for read access to logical rank 4 (CS 6). Refer to field 03:00 description for encodings.	0h	RW
23:20	R3ODTWR	**R3ODTWR**. Logical ODT vector for write access to logical rank 3 (CS 4). Refer to field 03:00 description for encodings.	0h	RW
19:16	R3ODTRD	**R3ODTRD**. Logical ODT vector for read access to logical rank 3 (CS 4). Refer to field 03:00 description for encodings.	0h	RW
15:12	R2ODTWR	**R2ODTWR**. Logical ODT vector for write access to logical rank 2 (CS 2). Refer to field 03:00 description for encodings.	0h	RW

Table 200. Offset B0 - B3h: DDR2ODTC – DDR2 ODT Control Register (Sheet 2 of 2)

Device: 0	**Function:** 0
Offset: B0 - B3h	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
11:08	R2ODTRD	**R2ODTRD.** Logical ODT vector for read access to logical rank 2 (CS 2). Refer to field 03:00 description for encodings.	0h	RW
07:04	R1ODTWR	**R1ODTWR**. Logical ODT vector for write access to logical rank 1 (CS 0). Refer to field 03:00 description for encodings.	0h	RW
03:00	R1ODTRD	**R1ODTRD.** Logical ODT vector for read access to logical rank 1 (CS 0). The four bits correspond to CS[7,5,3,1] values driven on the multi-function CS lines used for ODT in DDR2 mode. when the DRM register is set to 0x8421 which maps logical CS directly to physical CS. For each asserted bit in the nibble, the corresponding ODT output will be driven during read accesses to rank 1. So a value of 5h would cause logical CS5 and logical CS1 to be driven asserted, and logical CS7 and logical CS3 to remain de-asserted during reads to rank 1. The DDR2ODTC table in the DRAM Technology for DDR2 shows various settings for this register, based on DIMM topologies.	0h	RW

13.1.1.58 Offset C4 - C5h: TOLM – Top of Low Memory Register

This register contains the maximum address below 4 Gbyte that must be treated as a memory access and is defined on a 128 Mbyte boundary. Usually it is below the areas configured for PCI Express, NSI, and PCI memory. The memory address found in DRB7 reflects the amount of total memory populated. In the event that the combined DRAM and PCI memory space in the system is less than 4 Gbyte, these two registers are identical.

Table 201. Offset C4 - C5h: TOLM – Top of Low Memory Register

Device: 0	**Function:** 0
Offset: C4 - C5h	**Size:** 16 bit
Default Value: 0800h	

Bits	Name	Description	Reset Value	Access
15:11	TOLM	**Top of Low Memory:** This register corresponds to bits 31 to 27 of the system address which is 1 greater than the maximum DRAM location below 4 Gbyte. Configuration software must set this value to either the maximum amount of memory in the system or to the minimum address allocated for PCI memory or the graphics aperture, whichever is smaller. Address bits 26:00 are assumed to be 0 for the purposes of address comparison. Addresses equal to or greater than the TOLM, and less than 4 G, are treated as non-memory accesses. All accesses less than the TOLM are treated as DRAM accesses (except for the 15–16 Mbyte or PAM gaps). This register must be set to at least 0800h, for a minimum of 128 Mbyte of DRAM. There is also a minimum of 128 Mbyte of PCI space, since this register is on a 128 Mbyte boundary. Configuration software must set this value to either the maximum amount of memory in the system (same as DRB7), or to the lower 128 Mbyte boundary of the Memory Mapped IO range, whichever is smaller. Programming example: 1100_0b = 3 Gbyte (assuming that DBR7 is set > 4 Gbyte): An access to 0_C000_0000h or above (but <4 Gbyte) is considered above the TOLM and therefore not to DRAM. It may go to one of the PEA ports or NSI or be subtracted and decoded to NSI. An access to 0_BFFF_FFFFh and below is considered below the TOLM and go to DRAM.	00001b	RW
10:00	Reserved	Reserved	000h	

13.1.1.59 Offset C6 - C7h: REMAPBASE – Remap Base Address Register

Table 202. Offset C6 - C7h: REMAPBASE – Remap Base Address Register

Device: 0	**Function:** 0
Offset: C6 - C7h	**Size:** 16 bit
Default Value: 03FFh	

Bits	Name	Description	Reset Value	Access
15:10	Reserved	Reserved	00h	
09:00	REMAPBASE	**Remap Base Address Bits [35:26]:** The value in this register defines the lower boundary of the Remap window. The Remap window is inclusive of this address. In the decoder A[25:0] of the Remap Base Address are assumed to be 0s. Thus the bottom of the defined memory range is aligned to a 64 Mbyte boundary. When the value in this register is greater than the value programmed into the Remap Limit register, the Remap window is disabled. This field defaults to 3FFh.	3FFh	RW

13.1.1.60 Offset: C8 - C9h: REMAPLIMIT – Remap Limit Address Register

Table 203. Offset: C8 - C9h: REMAPLIMIT – Remap Limit Address Register

Device: 0	***Function:*** 0
Offset: *C8 - C9h*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:10	Reserved	Reserved	00h	
09:00	REMAPLMT	**Remap Limit Address Bits [35:26]:** The value in this register defines the upper boundary of the Remap window. The Remap window is inclusive of this address. In the decoder A[25:00] of the Remap Limit Address are assumed to be Fs. Thus the top of the defined range is one less than a 64 Mbyte boundary. When the value in this register is less than the value programmed into the Remap Base register, the Remap window is disabled.	00h	RW

13.1.1.61 Offset CA - CBh: REMAPOFFSET – Remap Offset Register

This register contains the difference between the REMAPBASE and TOLM.

Table 204. Offset CA - CBh: REMAPOFFSET – Remap Offset Register

Device: 0	***Function:*** 0
Offset: *CA - CBh*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:10	Reserved	Reserved	00h	
09:00	REMAPOFFST	**Remap Offset:** This register contains the difference between the REMAPBASE and TOLM. This register value corresponds to address bits 35:26. It is used to translate the physical FSB address to the system memory address for accesses to the remap region.	000h	RW

13.1.1.62 Offset CC - CDh: TOM – Top Of Memory Register

This register contains the effective size of memory. The value in this register hides any DIMMs that can't be directly addressed. BIOS determines the memory size reported to the OS using this register.

Table 205. Offset CC - CDh: TOM – Top Of Memory Register

Device: 0	***Function:*** 0
Offset: *CC - CDh*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:09	Reserved	Reserved	00h	
08:00	TOM	**Top of Memory:** This register reflects the effective size of memory. These bits correspond to address bits 35:27. (128 Mbyte granularity) Bits 26:00 are assumed to be 0.	000h	RW

13.1.1.63 Offset CE - CFh: HECBASE – PCI Express Port A (PEA) Enhanced Configuration Base Address Register

Configuration software reads this register to determine where the 256 Mbyte range of addresses resides for this particular host bridge. This register contains the base address of enhanced configuration memory.

Table 206. Offset CE - CFh: HECBASE – PCI Express Port A (PEA) Enhanced Configuration Base Address Register

Device: 0	***Function:*** 0
Offset: *CE - CFh*	***Size:*** *16 bit*
Default Value: *E000h*	

Bits	Name	Description	Reset Value	Access
15:12	HECBASE	**PEA Enhanced Configuration Base:** This register contains the address that corresponds to bits 31 to 28 of the base address for PEA enhanced configuration space below 4 Gbyte. Configuration software reads this register to determine where the 256 Mbyte range of addresses resides for this particular host bridge. BIOS needs to write this register at boot time. Settings 0 and F are not valid. When any byte or combination of bytes of this register is written, the register value locks down and cannot be further updated.	1110b	RWO
11:00	Reserved	Reserved.	000h	

13.1.1.64 Offset D8h: CACHECTL0 – Write Cache Control 0 Register

Table 207. Offset D8h: CACHECTL0 – Write Cache Control 0 Register

Device: 0	**Function:** 0
Offset: D8h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07:01	Reserved	Reserved	00h	
00	WCFLUSH	**Write Cache Flush:** 0 = Cleared by hardware when flush is complete. 1 = All entries in the write cache are flushed to DRAM with high priority. The arbiter no longer accepts requests until the write cache has been flushed. Software can poll this bit to determine when the flush is complete.	0b	RWS

13.1.1.65 Offset DE - DFh: SKPD – Scratchpad Data Register

Table 208. Offset DE - DFh: SKPD – Scratchpad Data Register

Device: 0	**Function:** 0
Offset: DE - DFh	**Size:** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
15:00	SCRTCH	**Scratchpad:** These bits are simply Read/Write storage bits that have no effect on the IMCH functionality. BIOS typically programs this register to the revision ID of the Memory Reference Code.	0000h	RW

13.1.1.66 Offset F5h: IMCH TST1 – IMCH Test Byte 1 Register

Table 209. Offset F5h: IMCH TST1 – IMCH Test Byte 1 Register

Device: 0	**Function:** 0
Offset: F5h	**Size:** 8 bit
Default Value: 01h	

Bits	Name	Description	Reset Value	Access
07:01	Reserved	Reserved	00h	RO
00	DCME	Disable Compliance Mode Entry. [STICKY]: 0 = Compliance Mode entry is enabled on all PCI Express ports. This is only set to '0' if doing compliance testing. 1 = When this bit is set to a 1, this prevents entry into Compliance Mode for all PEA ports. Software must ensure this bit set to '1' for normal operation.	1b	RW

13.2 Device 0, Function 1: DRAM Controller Error Reporting Registers

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 210. Error Reporting PCI Configuration Register Map (D0, F1) (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	01h	VID	Vendor Identification Register	8086h	RO
02h	03h	DID	Device Identification Register	35B1h	RO
04h	05h	PCICMD	PCI Command Register	0000h	RO, RW
06h	07h	PCISTS	PCI Status Register	0000h	RO, RWC
08h	08h	RID	Revision Identification Register	00h	RO
0Ah	0Ah	SUBC	Sub-Class Code Register	00h	RO
0Bh	0Bh	BCC	Base Class Code Register	FFh	RO
0Dh	0Dh	MLT	Master Latency Timer Register	00h	RO
0Eh	0Eh	HDR	Header Type Register	00h	RO
2Ch	2Dh	SVID	Subsystem Vendor Identification Register	0000h	RWO
2Eh	2Fh	SID	Subsystem Identification Register	0000h	RWO
40h	43h	GLOBAL_FERR	Global First Error Register	0000_0000h	RO, RWC
44h	47h	GLOBAL_NERR	Global Next Error Register	0000_0000h	RO, RWC
48h	4Bh	NSI_FERR	NSI First Error Register	0000_0000h	RO, RWC
4Ch	4Fh	NSI_NERR	NSI Next Error Register	0000_0000h	RO, RWC
50h	53h	NSI_SCICMD	NSI SCI Command Register	0000_0000h	RO, RW
54h	57h	NSI_SMICMD	NSI SMI Command Register	0000_0000h	RO, RW
58h	5Bh	NSI_SERRCMD	NSI SERR Command Register	0000_0000h	RO, RW
5Ch	5Fh	NSI_MCERRCMD	NSI MCERR Command Register	0000_0000h	RO, RW
60h	61h	FSB_FERR	FSB First Error Register	0000h	RO, RWC
62h	63h	FSB_NERR	FSB Next Error Register	0000h	RO, RWC
64h	65h	FSB_EMASK	FSB Error Mask Register	0009h	RO, RW
68h	69h	FSB_SCICMD	FSB SCI Command Register	0000h	RO, RW
6Ah	6Bh	FSB_SMICMD	FSB SMI Command Register	0000h	RO, RW
6Ch	6Dh	FSB_SERRCMD	FSB SERR Command Register	0000h	RO, RW
6Eh	6Fh	FSB_MCERRCMD	FSB MCERR Command Register	0000h	RO, RW
70h	70h	BUF_FERR	Memory Buffer First Error Register	00h	RO, RWC
72h	72h	BUF_NERR	Memory Buffer Next Error Register	00h	RO, RWC
74h	74h	BUF_EMASK	Memory Buffer Error Mask Register	00h	RO, RW
78h	78h	BUF_SCICMD	Memory Buffer SCI Command Register	00h	RO, RW
7Ah	7Ah	BUF_SMICMD	Memory Buffer SMI Command Register	00h	RO, RW
7Ch	7Ch	BUF_SERRCMD	Memory Buffer SERR Command Register	00h	RO, RW
7Eh	7Eh	BUF_MCERRCMD	Memory Buffer MCERR Command Register	00h	RO, RW
80h	81h	DRAM_FERR	DRAM First Error Register	0000h	RWC
82h	83h	DRAM_NERR	DRAM Next Error Register	0000h	RWC
84h	84h	DRAM_EMASK	DRAM Error Mask Register	00h	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 210. Error Reporting PCI Configuration Register Map (D0, F1) (Sheet 2 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
88h	88h	DRAM_SCICMD	DRAM SCI Command Register	00h	RW
8Ah	8Ah	DRAM_SMICMD	DRAM SMI Command Register	00h	RW
8Ch	8Ch	DRAM_SERRCMD	DRAM SERR Command Register	00h	RW
8Eh	8Eh	DRAM_MCERRCMD	DRAM MCERR Command Register	00h	RW
90h	93h	NSI_EMASK	NSI Error Mask Register	0000_0000h	RO, RW
94h	97h	NSI_ERRSID	NSI Error Message Source ID Register	0000_0000h	RO
98h	99h	THRESH_SEC0	DIMM0 SEC Threshold Register	0000h	RW
9Ah	9Bh	THRESH_SEC1	DIMM1 SEC Threshold Register	0000h	RW
9Ch	9Dh	THRESH_SEC2	DIMM2 SEC Threshold Register	0000h	RW
9Eh	9Fh	THRESH_SEC3	DIMM3 SEC Threshold Register	0000h	RW
A0h	A3h	DRAM_SECF_ADD	DRAM First Single Bit Error Correct Address Register	0000_0000h	RO
A4h	A7h	DRAM_DED_ADD	DRAM Double Bit Error Address Register	0000_0000h	RO
A8h	ABh	DRAM_SCRB_ADD	DRAM Scrub Error Address Register	0000_0000h	RO
ACh	AFh	DRAM_RETR_ADD	DRAM DED Retry Address Register	0000_0000h	RO
B0h	B1h	DRAM_SEC_D0A	DRAM SEC Logical DIMM 0 Counter Register	0000h	RW
B2h	B3h	DRAM_DED_D0A	DRAM DED Logical DIMM 0 Counter Register	0000h	RW
B4h	B5h	DRAM_SEC_D1A	DRAM SEC Logical DIMM 1 Counter Register	0000h	RW
B6h	B7h	DRAM_DED_D1A	DRAM DED Logical DIMM 1 Counter Register	0000h	RW
B8h	B9h	DRAM_SEC_D2A	DRAM SEC Logical DIMM 2 Counter Register	0000h	RW
BAh	BBh	DRAM_DED_D2A	DRAM DED Logical DIMM 2 Counter Register	0000h	RW
BCh	BDh	DRAM_SEC_D3A	DRAM SEC Logical DIMM 3 Counter Register	0000h	RW
BEh	BFh	DRAM_DED_D3A	DRAM DED Logical DIMM 3 Counter Register	0000h	RW
C2h	C3h	THRESH_DED	Threshold for DEDs Register	0000h	RW
C4h	C5h	DRAM_SECF_ SYNDROME	DRAM First Single Error Correct Syndrome Register	0000h	RO
C6h	C7h	DRAM_SECN_ SYNDROME	DRAM Next Single Error Correct Syndrome Register	0000h	RO
C8h	CBh	DRAM_SECN_ADD	DRAM Next Single Bit Error Correct Address Register	0000_0000h	RO
DCh	DDh	DIMMTHREX	DIMM Threshold Exceeded Register	0000h	RWC
E0h	E3h	HERRCTL	Host Error Control Register	0020_0000h	RO, RW
E8h	EBh	BERRCTL	Buffer Error Control Register	0000_0000h	RO, RW
ECh	EFh	DERRCTL	DRAM Error Control Register	0000_0000h	RO, RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

13.2.1 Register Details

13.2.1.1 Offset 00 - 01h: VID – Vendor Identification Register

The VID Register contains the vendor identification number. This 16-bit register combined with the Device Identification Register uniquely identify any PCI device.

Table 211. Offset 00 - 01h: VID – Vendor Identification Register

Device: 0 ***Function:*** 1
Offset: *00 - 01h* ***Size:*** *16 bit*
Default Value: *8086h*

Bits	Name	Description	Reset Value	Access
15:00	VID	**Vendor Identification:** This register field contains the PCI standard identification for Intel 8086h.	8086h	RO

13.2.1.2 Offset 02 - 03h: DID – Device Identification Register

This 16-bit register combined with the Vendor Identification register uniquely identifies any PCI device.

Table 212. Offset 02 - 03h: DID – Device Identification Register

Device: 0 ***Function:*** 1
Offset: *02 - 03h* ***Size:*** *16 bit*
Default Value: *35B1h*

Bits	Name	Description	Reset Value	Access
15:00	DID	**Device Identification Number:** This is a 16-bit value assigned to the IMCH Host-Bridge Function 1.	35B1h	RO

13.2.1.3 Offset 04 - 05h: PCICMD – PCI Command Register

Table 213. Offset 04 - 05h: PCICMD – PCI Command Register

Device: 0 ***Function:*** 1
Offset: *04 - 05h* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:09	Reserved	Reserved	00h	
08	SERRE	**SERR Enable:** This bit is a global enable bit for Device 0 SERR messaging. 0 = Disable, SERR message is not generated by the IMCH for Device 0, Function 1. 1 = The IMCH is enabled to generate SERR messages over the NSI for specific Device 0, Function 1 error conditions that are individually enabled in the ERRCMD registers. The IMCH communicates the SERR condition by sending a DO_SERR message over NSI to the IICH. If this bit is set to a 1, the IMCH is enabled to generate SERR messages over NSI for specific Device 0, Function 1 error conditions that are individually enabled in the NSI_SERRCMD, FSB_SERRCMD, BUF_SERRCMD, and DRAM_SERRCMD registers. The error status is reported in the PCISTS register as well as the corresponding FERR/NERR registers. ***Note:*** Reporting via SERR for detected parity error which is essentially NSI Poisoned TLP's, can ALSO be reported through by the Device 0, Function 0.	0b	RW
07:00	Reserved	Reserved	00h	

13.2.1.4 Offset 06 - 07h: PCISTS – PCI Status Register

PCISTS is a 16-bit status register that reports the occurrence of error events on Device 0's PCI interface.

Table 214. Offset 06 - 07h: PCISTS – PCI Status Register

Device: 0 ***Function:*** 1
Offset: *06 - 07h* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15	Reserved	Reserved	0b	
14	SSE	**Signaled System Error:** 0 = SERR is not generated by IMCH Device 0, Function 1 1 = IMCH Device 0 Function 1 generated a SERR message over NSI for any enabled Device 0, Function 1 error condition. Software clears this bit by writing a 1 to the bit location.	0b	RWC
13:00	Reserved	Reserved	000h	

13.2.1.5 Offset 08h: RID – Revision Identification Register

This register contains the revision number of the IMCH Device 0.

Table 215. Offset 08h: RID – Revision Identification Register

Device: 0	**Function:** 1
Offset: 08h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07:00	RID	**Revision Identification Number:** This value indicates the revision identification number for the IMCH Device 0. This number must always be the same as the RID for function 0.	00h	RO

13.2.1.6 Offset 0Ah: SUBC – Sub-Class Code Register

Table 216. Offset 0Ah: SUBC – Sub-Class Code Register

Device: 0	**Function:** 1
Offset: 0Ah	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07:00	SUBC	**Sub-Class Code:** This value indicates the Sub Class Code into which the IMCHDevice 0, Function 1 falls. 00h = Bridge	00h	RO

13.2.1.7 Offset 0Bh: BCC – Base Class Code Register

Table 217. Offset 0Bh: BCC – Base Class Code Register

Device: 0	**Function:** 1
Offset: 0Bh	**Size:** 8 bit
Default Value: FFh	

Bits	Name	Description	Reset Value	Access
07:00	BASEC	**Base Class Code:** This value indicates the Base Class Code for the IMCH Device 0, Function 1. FFh = A 'non-defined' device. Since this function is used for error conditions, it does not fall into any other class.	FFh	RO

13.2.1.8 Offset 0Eh: HDR – Header Type Register

Table 218. Offset 0Eh: HDR – Header Type Register

Device: 0	***Function:*** 1
Offset: *0Eh*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:00	HDR	**PCI Header:** This value indicates the Header Type for the IMCH Device 0. 00h = IMCH is a multi-function device with a standard header layout.	00h	RO

13.2.1.9 Offset 2C - 2Dh: SVID – Subsystem Vendor Identification Register

This value is used to identify the vendor of the subsystem.

Table 219. Offset 2C - 2Dh: SVID – Subsystem Vendor Identification Register

Device: 0	***Function:*** 1
Offset: *2C - 2D*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:00	SUBVID	**Subsystem Vendor ID:** This field must be programmed during boot-up to indicate the vendor of the system board.	0000h	RWO

13.2.1.10 Offset 2E - 2Fh: SID – Subsystem Identification Register

This value is used to identify a particular subsystem.

Table 220. Offset 2E - 2Fh: SID – Subsystem Identification Register

Device: 0	***Function:*** 1
Offset: *2E - 2F*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:00	SUBID	**Subsystem ID:** This field must be programmed during BIOS initialization.	0000h	RWO

13.2.1.11 Offset 40 - 43h: GLOBAL_FERR – Global First Error Register

This register is used to log various error conditions at the “unit” level. These bits are “sticky” through reset, and are set regardless of whether or not any error messages (SCI, SMI, SERR#, MCERR#) are enabled and generated at the unit level. Specific error conditions within the various functional units are logged in the unit-specific error registers that follow.

This register captures the FIRST global Fatal and the FIRST global Non-Fatal errors. For these global error registers, a non-fatal error can be either an uncorrectable error which is non-fatal, or a correctable error. Any future errors (NEXT errors) are captured in the NERR_Global register. No further error bits in this register are set until the existing error bit is cleared.

Note: If multiple errors are reported in the same clock as the first error, all errors are latched.

Table 221. Offset 40 - 43h: GLOBAL_FERR – Global First Error Register (Sheet 1 of 2)

Device: 0 **Function:** 1
Offset: 40 - 43h **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:28	Reserved	Reserved	00h	
27	DRAM_FE	**DRAM Controller Channel Fatal Error:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No fatal DRAM I/F error. 1 = The IMCH detected a fatal DRAM interface error.	0b	RWC
26	FSB_FE	**Host (FSB) Fatal Error:**This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No fatal FSB error. 1 = The IMCH detected a fatal FSB error.	0b	RWC
25	NSI_FE	NSI **Fatal Error:**This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No fatal NSI error. 1 = The IMCH detected a fatal NSI error.	0b	RWC
24	DMA_FE	**DMA Controller Fatal Error Device 1 fatal error (EDMA):** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No fatal EDMA Controller error. 1 = The IMCH detected a fatal EDMA controller error.	0b	RWC
23	PA_FE	**PCI Express* Port A0 Fatal Error:**This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No fatal PCI Express Port A error. 1 = The IMCH detected a fatal PCI Express Port A0 error.	0b	RWC
22	PA1_FE	**PCI Express Port A1 Fatal Error** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No fatal PCI Express Port A1 error. 1 = The IMCH detected a fatal PCI Express Port A1 error.	0b	RWC
21:15	Reserved	Reserved	00h	
14	BUFF_NFE	**Buffer unit detected non-fatal error:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No non-fatal Buffer error. 1 = The IMCH detected a non-fatal Buffer error.	0b	RWC
13	DRAM_NFE	**DRAM Controller Non-Fatal Error:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No non-fatal DRAM Controller error. 1 = The IMCH detected a non-fatal DRAM Controller error.	0b	RWC
12	FSB_NFE	**FSB Non-Fatal Error:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No non-fatal FSB error. 1 = The IMCH detected a non-fatal FSB error.	0b	RWC

Table 221. Offset 40 - 43h: GLOBAL_FERR – Global First Error Register (Sheet 2 of 2)

Device: 0		**Function:** 1		
Offset: 40 - 43h		**Size:** 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
11	NSI_NFE	NSI **Non-Fatal Error:**This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No non-fatal NSI error. 1 = The IMCH detected a non-fatal NSI error.	0b	RWC
10	DMA_NFE	**DMA Controller Non-Fatal Error Device 1 non-fatal error (EDMA):** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No non-fatal EDMA Controller error. 1 = The IMCH detected a non-fatal EDMA Controller error.	0b	
09	PA_NFE	**PCI Express Port A0 Non-Fatal Error:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No non-fatal PCI Express Port A error. 1 = The IMCH detected a non-fatal PCI Express Port A error.	0b	RWC
08	PA1_NFE	**PCI Express Port A1 Non-Fatal Error:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No non-fatal PCI Express Port A1 error. 1 = The IMCH detected a non-fatal PCI Express Port A1 error.	0b	RWC
07:00	Reserved	Reserved	0oh	

13.2.1.12 Offset 44 - 47h: GLOBAL_NERR – Global Next Error Register

The bit definitions are the same as defined for FERR_GLOBAL.

Table 222. Offset 44 - 47h: GLOBAL_NERR – Global Next Error Register

Device: 0		**Function:** 1		
Offset: 44 - 47h		**Size:** 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.2.1.11, "Offset 40 - 43h: GLOBAL_FERR – Global First Error Register" for details.				

13.2.1.13 Offset 48 - 4Bh: NSI_FERR – NSI First Error Register

NSI errors for NSI port to IICH. These errors include errors detected on the NSI link, errors from the NSI hierarchy, and errors internal to the NSI unit.

Note: If two errors occur within 1 clock cycle of each other in the NSI, the first error will get logged into FERR but the error on the next clock edge may or may not get logged into the NERR. All follow on errors after the second cycle will get captured into the NERR

Table 223. Offset 48 - 4Bh: NSI_FERR – NSI First Error Register (Sheet 1 of 3)

Device: 0	***Function:*** 1
Offset: *48 - 4Bh*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
31:30	Reserved	Reserved	00b	
29	UR	**Unsupported Request[STICKY]:** This bit is sticky through reset. 0 = Cleared by writing a '1' to the bit location. 1 = Unsupported Request detected.(NON-FATAL.)	0b	RWC
28	Reserved	Reserved	0b	
27	MTLP	**Malformed TLP Status [STICKY]:** Malformed TLP errors include: data payload length issues, byte enable rule violations, and various other illegal field settings. This bit is sticky through reset. 0 = Cleared by writing a '1' to the bit location. 1 = Malformed TLP detected.(FATAL.)	0b	RWC
26	ROVF	**Receiver Overflow Status [STICKY]:** IMCH checks for overflows on the following upstream queues: posted, non-posted, and completion. This bit is sticky through reset. 0 = Cleared by writing a '1' to the bit location. 1 = Receiver Overflow detected.(FATAL)	0b	RWC
25	UEC	**Unexpected Completion Status [STICKY]:** This bit is set when the device receives a completion which does not correspond to any of the outstanding requests issued by that device. This bit is sticky through reset. 0 = Cleared by writing a '1' to the bit location. 1 = Unexpected Completion detected.(NON-FATAL)	0b	RWC
24	CA	**Completer Abort Status [STICKY]:** If a request received violates the specific programming model of this device, but is otherwise legal, this bit is set. This bit is sticky through reset. 0 = Cleared by writing a '1' to the bit location. 1 = Completer Abort detected.(NON-FATAL)	0b	RWC
23	CT	**Completion Timeout Status [STICKY]:** The Completion Timeout timer must expire if a Request is not completed in 50 ms, but must not expire earlier than 50 µs. When the timer expires, this bit is set. This bit is sticky through reset. 0 = Cleared by writing a '1' to the bit location. 1 = Completion timeout detected.(NON-FATAL)	0b	RWC
22	Reserved	Reserved	0b	
21	PTLP	**Poisoned TLP Status [STICKY]:** This bit when set indicates that some portion of the TLP data payload was corrupt. This bit is sticky through reset. 0 = Cleared by writing a '1' to the bit location. 1 = Poisoned TLP detected.(NON-FATAL)	0b	RWC
20	Reserved	Reserved	0b	

Table 223. Offset 48 - 4Bh: NSI_FERR – NSI First Error Register (Sheet 2 of 3)

Device: 0 **Function:** 1
Offset: 48 - 4Bh **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
19	DLPE	**Data Link Protocol Error Status [STICKY]:**This bit is set when an ACK/NAK received does not specify the sequence number of an unacknowledged TLP, or of the most recently acknowledged TLP. This bit is sticky through reset. 0 = Cleared by writing a '1' to the bit location. 1 = Data Link Protocol Error detected.(FATAL)	0b	RWC
18:16	Reserved	Reserved	0b	
15	RTTO	**Replay Timer Timeout Status [STICKY]:** The replay timer counts time since the last ACK or NAK DLLP was received. When the timer expires, this bit is set. This bit is sticky through system reset. 0 = Cleared by writing a '1' to the bit location. 1 = Replay Timer timeout detected.(NON-FATAL)	0b	RWC
14	Reserved	Reserved	0b	
13	RNRO	**REPLAY_NUM Rollover Status [STICKY]:** A 2-bit counter counts the number of times the retry buffer has been retransmitted. When this counter rolls over, this bit is set. This bit is sticky through system reset. 0 = Cleared by writing a '1' to the bit location. 1 = REPLAY_NUM rollover detected.(NON-FATAL)	0b	RWC
12	BDLLP	**Bad DLLP Status [STICKY]:** This bit is set when the calculated DLLP CRC is not equal to the received value. Also included are 8b/10b errors within the TLP including wrong disparity. An invalid sequence number also sets this bit. This bit is sticky through system reset. 0 = Cleared by writing a '1' to the bit location. 1 = Bad DLLP detected.(NON-FATAL)	0b	RWC
11	BTLP	**Bad TLP Status [STICKY]:** 0 = The calculated TLP CRC is equal to the received value. 1 = The calculated TLP CRC is not equal to the received value. Also included are 8b/10b errors within the TLP including wrong disparity, and invalid sequence numbers.(NON-FATAL)	0b	RWC
10	Reserved	Reserved	0b	
09	RCVRE	**Receiver Error Status [STICKY]:** Data is delivered over PCI Express via packets built out of 8b/10b symbols. This error is set for problems with the packet framing around these symbols or with symbols received outside of recognized packets. This bit is sticky through system reset. 0 = Cleared by writing a '1' to the bit location. 1 = Receiver Error detected. ***Note:*** (NON-FATAL)	0b	RWC
08	Reserved	Reserved	0b	
07	FEMR	**Fatal Error Message Received:** 0 = No Fatal Error Message Received over the NSI link. 1 = Fatal Error Message Received over the NSI link.(FATAL)	0b	RWC
06	NEMR	**Non-Fatal Error Message Received:** Non-Fatal Error Message Received over the NSI link. 0 = No Non-Fatal Error Message Received over the NSI link. 1 = Non-Fatal Error Message Received over the NSI link.(NON-FATAL)	0b	RWC

Table 223. Offset 48 - 4Bh: NSI_FERR – NSI First Error Register (Sheet 3 of 3)

Device: 0	**Function:** 1
Offset: 48 - 4Bh	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
05	CEMR	**Correctable Error Message Received:** Correctable Error Message Received over the NSI link. 0 = No Correctable Error Message Received over the NSI link. 1 = Correctable Error Message Received over the NSI link.(NON-FATAL)	0b	RWC
04:03	Reserved	Reserved	00b	
02	PED	**Parity Error Detected during parity conversion from CTB:** Parity Error detected on data received from the core. 0 = No Parity Error detected on data received from the core. 1 = Parity Error detected on data received from the core.(NON-FATAL)	0b	RWC
01	Reserved	Reserved	0b	
00	LD	**Link Down:** 0 = Link has not transitioned from DL_UP to DL_DOWN. 1 = Link transitioned from DL_UP to DL_DOWN. (FATAL)	0b	RWC

13.2.1.14 Offset 4C - 4Fh: NSI_NERR – NSI Next Error Register

Errors that are detected after the first error are captured by this register.

Table 224. Offset 4C - 4Fh: NSI_NERR – NSI Next Error Register

Device: 0	**Function:** 1
Offset: 4C - 4Fh	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
See Section 13.2.1.13, "Offset 48 - 4Bh: NSI_FERR – NSI First Error Register" on page 352 for bit definitions.				

13.2.1.15 Offset 50 - 53h: NSI_SCICMD – NSI SCI Command Register

This register enables various errors to generate an SCI special cycle. When an error flag is set in the FERR or NERR registers, it can generate an SCI special cycle when enabled in the SCICMD registers. Note that one and only one message type can be enabled.

Table 225. Offset 50 - 53h: NSI_SCICMD – NSI SCI Command Register (Sheet 1 of 2)

Device: 0 **Function:** 1
Offset: 50 - 53h **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:30	Reserved	Reserved	00b	
29	UR_SCI	**Generate SCI for** NSI **Error 29:** Generate SCI whenever bit 29 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
28	Reserved	Reserved	0b	
27	MTLP_SCI	**Generate SCI for** NSI **Error 27:** Generate SCI whenever bit 27 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
26	ROVF_SCI	**Generate SCI for** NSI **Error 26:** Generate SCI whenever bit 26 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
25	UEC_SCI	**Generate SCI for** NSI **Error 25:** Generate SCI whenever bit 25 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
24	CA_SCI	**Generate SCI for** NSI **Error 24:** Generate SCI whenever bit 24 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
23	CT_SCI	**Generate SCI for** NSI **Error 23:** Generate SCI whenever bit 23 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
22	Reserved	Reserved	0b	
21	PTLP_SCI	**Generate SCI for** NSI **Error 21:** Generate SCI whenever bit 21 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
20	Reserved	Reserved	0b	
19	DLPE_SCI	**Generate SCI for** NSI **Error 19:** Generate SCI whenever bit 19 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
18:16	Reserved	Reserved	000b	
15	RTTO_SCI	**Generate SCI for** NSI **Error 15:** Generate SCI whenever bit 15 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
14	Reserved	Reserved	0b	
13	RNRO_SCI	**Generate SCI for** NSI **Error 13:** Generate SCI whenever bit 13 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
12	BDLLP_SCI	**Generate SCI for** NSI **Error 12:** Generate SCI whenever bit 12 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW

Table 225. Offset 50 - 53h: NSI_SCICMD – NSI SCI Command Register (Sheet 2 of 2)

Device: 0	***Function:*** 1
Offset: *50 - 53h*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
11	BTLP_SCI	**Generate SCI for** NSI **Error 11:** Generate SCI whenever bit 11 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
10	Reserved	Reserved	0b	
09	RCVRE_SCI	**Generate SCI for** NSI **Error 9:** Generate SCI whenever bit 9 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
08	Reserved	Reserved	0b	
07	FEMR_SCI	**Generate SCI for** NSI **Error 7:** Generate SCI whenever bit 7 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
06	NEMR_SCI	**Generate SCI for** NSI **Error 6:** Generate SCI whenever bit 6 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
05	CEMR_SCI	**Generate SCI for** NSI **Error 5:** Generate SCI whenever bit 5 of NSI _FERR or NSI _NERR is set 0 = Disable 1 = Enable	0b	RW
04:03	Reserved	Reserved	00b	
02	PED_SCI	**Generate SCI for** NSI **Error 2:** Generate SCI whenever bit 2 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
01	Reserved	Reserved	0b	
00	LD_SCI	**Generate SCI for** NSI **Error 0:** Generate SCI whenever bit 0 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW

13.2.1.16 Offset 54 - 57h: NSI_SMICMD – NSI SMI Command Register

This register enables various errors to generate an SMI NSI special cycle. When an error flag is set in the FERR or NERR registers it generates an SMI NSI special cycle when enabled in the NSI_SMICMD register. Note that one and only one message type can be enabled.

Table 226. Offset 54 - 57h: NSI_SMICMD – NSI SMI Command Register (Sheet 1 of 2)

Device: 0 **Function:** 1
Offset: 54 - 57h **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:30	Reserved	Reserved	00b	
29	UR_SMI	**Generate SMI for** NSI **Error 29:** Generate SMI whenever bit 29 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
28	Reserved	Reserved	0b	
27	MTLP_SMI	**Generate SMI** NSI **for Error 27:** Generate SMI whenever bit 27 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
26	ROVF_SMI	**Generate SMI for** NSI **Error 26:** Generate SMI whenever bit 26 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
25	UEC_SMI	**Generate SMI for** NSI **Error 25:** Generate SMI whenever bit 25 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
24	CA_SMI	**Generate SMI for** NSI **Error 24:** Generate SMI whenever bit 24 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
23	CT_SMI	**Generate SMI for** NSI **Error 23:** Generate SMI whenever bit 23 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
22	Reserved	Reserved	0b	
21	PTLP_SMI	**Generate SMI for** NSI **Error 21:** Generate SMI whenever bit 21 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
20	Reserved	Reserved	0b	
19	DLPE_SMI	**Generate SMI for** NSI **Error 19:** Generate SMI whenever bit 19 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
18:16	Reserved	Reserved	000b	
15	RTTO_SMI	**Generate SMI for** NSI **Error 15:** Generate SMI whenever bit 15 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
14	Reserved	Reserved	0b	
13	RNRO_SMI	**Generate SMI for** NSI **Error 13:** Generate SMI whenever bit 13 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
12	BDLLP_SMI	**Generate SMI for** NSI **Error 12:** Generate SMI whenever bit 12 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW

Table 226. Offset 54 - 57h: NSI_SMICMD – NSI SMI Command Register (Sheet 2 of 2)

Device: 0	***Function:*** 1
Offset: *54 - 57h*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
11	BTLP_SMI	**Generate SMI for** NSI **Error 11:** Generate SMI whenever bit 11 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
10	Reserved	Reserved	0b	
09	RCVRE_SMI	**Generate SMI for** NSI **Error 9:** Generate SMI whenever bit 9 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
08	Reserved	Reserved	0b	
07	FEMR_SMI	**Generate SMI for** NSI **Error 7:** Generate SMI whenever bit 6 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
06	NEMR_SMI	**Generate SMI for** NSI **Error 6:** Generate SMI whenever bit 6 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
05	CEMR_SMI	**Generate SMI for** NSI **Error 5:** Generate SMI whenever bit 5 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
04:03	Reserved	Reserved	00b	
02	PED_SMI	**Generate SMI for** NSI **Error 2:** Generate SMI whenever bit 2 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
01	Reserved	Reserved	0b	
00	LD_SMI	**Generate SMI for** NSI **Error 0:** Generate SMI whenever bit 0 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW

13.2.1.17 Offset 58 - 5Bh: NSI_SERRCMD – NSI SERR Command Register

This register enables various errors to generate an SERR NSI special cycle. When an error flag is set in the FERR or NERR registers it generates an SERR NSI special cycle when enabled in the SERRCMD register. Note that one and only one message type can be enabled.

Table 227. Offset 58 - 5Bh: NSI_SERRCMD – NSI SERR Command Register (Sheet 1 of 2)

Device: 0 **Function:** 1
Offset: 58 - 5Bh **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:30	Reserved	Reserved	00b	
29	UR_SERR	**Generate SERR for** NSI **Error 29:** Generate SERR whenever bit 29 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
28	Reserved	Reserved	0b	
27	MTLP_SERR	**Generate SERR for** NSI **Error 27:** Generate SERR whenever bit 27 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
26	ROVF_SERR	**Generate SERR for** NSI **Error 26:** Generate SERR whenever bit 26 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
25	UEC_SERR	**Generate SERR for** NSI **Error 25:** Generate SERR whenever bit 25 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
24	CA_SERR	**Generate SERR for** NSI **Error 24:** Generate SERR whenever bit 24 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
23	CT_SERR	**Generate SERR for** NSI **Error 23:** Generate SERR whenever bit 23 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
22	Reserved	Reserved	0b	
21	PTLP_SERR	**Generate SERR for** NSI **Error 21:** Generate SERR whenever bit 21 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
20	Reserved	Reserved	0b	
19	DLPE_SERR	**Generate SERR for** NSI **Error 19:** Generate SERR whenever bit 19 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
18:16	Reserved	Reserved	000b	
15	RTTO_SERR	**Generate SERR for** NSI **Error 15:** Generate SERR whenever bit 15 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
14	Reserved	Reserved	0b	
13	RNRO_SERR	**Generate SERR for** NSI **Error 13:** Generate SERR whenever bit 13 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
12	BDLLP_SERR	**Generate SERR for** NSI **Error 12:** Generate SERR whenever bit 12 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW

Table 227. Offset 58 - 5Bh: NSI_SERRCMD – NSI SERR Command Register (Sheet 2 of 2)

Device: 0	**Function:** 1
Offset: 58 - 5Bh	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
11	BTLP_SERR	**Generate SERR for** NSI **Error 11:** Generate SERR whenever bit 11 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
10	Reserved	Reserved	0b	
09	RCVRE_SERR	**Generate SERR for** NSI **Error 9:** Generate SERR whenever bit 9 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
08	Reserved	Reserved	0b	
07	FEMR_SERR	**Generate SERR for** NSI **Error 7:** Generate SERR whenever bit 7 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
06	NEMR_SERR	**Generate SERR for** NSI **Error 6:** Generate SERR whenever bit 6 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
05	CEMR_SERR	**Generate SERR for** NSI **Error 5:** Generate SERR whenever bit 5 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
04:03	Reserved	Reserved	00b	
02	PED_SERR	**Generate SERR for** NSI **Error 2:** Generate SERR whenever bit 2 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
01	Reserved	Reserved	0b	
00	LD_SERR	**Generate SERR for** NSI **Error 0:** Generate SERR whenever bit 0 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW

13.2.1.18 Offset 5C - 5Fh: NSI_MCERRCMD – NSI MCERR Command Register

This register enables various errors to generate the MCERR signal on the FSB. When an error flag is set in the FERR or NERR registers it generates a MCERR when enabled in the MCERRCMD.

Table 228. Offset 5C - 5Fh: NSI_MCERRCMD – NSI MCERR Command Register (Sheet 1 of 2)

Device: 0 **Function:** 1
Offset: 5C - 5Fh **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:30	Reserved	Reserved	0b	
29	UR_MCERR	**Generate MCERR for** NSI **Error 29:** Generate MCERR whenever bit 29 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
28	Reserved	Reserved	0b	
27	MTLP_MCERR	**Generate MCERR for** NSI **Error 27:** Generate MCERR whenever bit 27 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
26	ROVF_MCERR	**Generate MCERR for** NSI **Error 26:** Generate MCERR whenever bit 26 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
25	UEC_MCERR	**Generate MCERR for** NSI **Error 25:** Generate MCERR whenever bit 25 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
24	CA_MCERR	**Generate MCERR for** NSI **Error 24:** Generate MCERR whenever bit 24 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
23	CT_MCERR	**Generate MCERR for** NSI **Error 23:** Generate MCERR whenever bit 23 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
22	Reserved	Reserved	0b	
21	PTLP_MCERR	**Generate MCERR for** NSI **Error 21:** Generate MCERR whenever bit 21 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
20	Reserved	Reserved	0b	
19	DLPE_MCERR	**Generate MCERR for** NSI **Error 19:** Generate MCERR whenever bit 19 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
18:16	Reserved	Reserved	0b	
15	RTTO_MCERR	**Generate MCERR for** NSI **Error 15:** Generate MCERR whenever bit 15 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
14	Reserved	Reserved	0b	
13	RNRO_MCERR	**Generate MCERR for** NSI **Error 13:** Generate MCERR whenever bit 13 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
12	BDLLP_MCERR	**Generate MCERR for** NSI **Error 12:** Generate MCERR whenever bit 12 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW

Table 228. Offset 5C - 5Fh: NSI_MCERRCMD – NSI MCERR Command Register (Sheet 2 of 2)

Device: 0 **Function:** 1
Offset: 5C - 5Fh **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
11	BTLP_MCERR	**Generate MCERR for** NSI **Error 11:** Generate MCERR whenever bit 11 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
10	Reserved	Reserved	0b	
09	RCVRE_MCERR	**Generate MCERR for** NSI **Error 9:** Generate MCERR whenever bit 9 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
08	Reserved	Reserved	0b	
07	FEMR_MCERR	**Generate MCERR for** NSI **Error 7:** Generate MCERR whenever bit 7 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
06	NEMR_MCERR	**Generate MCERR for** NSI **Error 6:** Generate MCERR whenever bit 6 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
05	CEMR_MCERR	**Generate MCERR for** NSI **Error 5:** Generate MCERR whenever bit 5 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
04:03	Reserved	Reserved	0b	
02	PED_MCERR	**Generate MCERR for** NSI **Error 2:** Generate MCERR whenever bit 2 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW
01	Reserved	Reserved	0b	
00	LD_MCERR	**Generate MCERR for** NSI **Error 0:** Generate MCERR whenever bit 0 of NSI _FERR or NSI _NERR is set. 0 = Disable 1 = Enable	0b	RW

13.2.1.19 Offset 60 - 61h: FSB_FERR – FSB First Error Register

This register stores the first error related to the FSB. Only one error bit is set in this register. Any future errors (NEXT errors) are set in the FSB_NERR register. No further error bits in the FSB_FERR register are set until the existing error bit is cleared. These bits are sticky through reset. Software clears these bits by writing a 1 to the bit location.

Note: If multiple errors are reported in the same clock as the first error, all errors are latched.

Table 229. Offset 60 - 61h: FSB_FERR – FSB First Error Register (Sheet 1 of 2)

Device: 0 **Function:** 1
Offset: 60 - 61h **Size:** 16 bit
Default Value: 0000h

Bits	Name	Description	Reset Value	Access
15:10	Reserved	Reserved	00h	
09	PARIO	**Parity error on outgoing data from I/O:**. (NON-FATAL) 0 = No parity error detected. 1 = Parity error detected on parity conversion for outgoing data from I/O subsystem.	0b	RWC
08	PARM	**Parity error on outgoing data from memory:**. (NON-FATAL) 0 = No parity error detected. 1 = Parity error detected on parity conversion for outgoing data from memory.	0b	RWC
07	FSB_BINIT#	**FSB BINIT# detected:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No FSB BINIT# detected 1 = This bit is set on an electrical high-to-low transition (0 to 1) of BINIT#. (FATAL)	0b	RWC
06	FSB_MCERR#	**FSB MCERR# detected:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No FSB MCERR# detected 1 = This bit is set on an electrical high-to-low transition (0 to 1) of MCERR# when Intel® 3100 Chipset is not driving. (NON-FATAL)	0b	RWC
05	NDLOCK	**Non-DRAM Lock Error:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No DRAM Lock Error detected 1 = IMCH detected a lock operation to memory space that did not map into DRAM. (NON-FATAL)	0b	RWC
04	ATOM	**FSB Address Above TOM/TOLM:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No FSB address above TOM/TOLM detected 1 = IMCH has detected an address above the Top of Memory and above 4 Gbyte.If the system has less than 4 Gbyte of DRAM, then unclaimed addresses between TOLM and 4 Gbyte are sent to NSI. (NON-FATAL)	0b	RWC
03	FSBDPAR	**FSB Data Parity Error:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No FSB parity error detected. 1 = IMCH has detected a data parity error on the FSB. (NON-FATAL)	0b	RWC
02	FSBAGL	**FSB Address Strobe Glitch Detected:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No FSB address strobe glitch detected. 1 = IMCH has detected a glitch on one of the FSB address strobes. (FATAL)	0b	RWC
01	FSBDGL	**FSB Data Strobe Glitch Detected:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No FSB data strobe glitch detected. 1 = IMCH has detected a glitch one of the FSB data strobes. (FATAL)	0b	RWC

Table 229. Offset 60 - 61h: FSB_FERR – FSB First Error Register (Sheet 2 of 2)

Device: 0	**Function:** 1
Offset: 60 - 61h	**Size:** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
00	FSBRPAR	**FSB Request/Address parity Error:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No FSB request/address parity error detected. 1 = IMCH has detected a parity error on either the address or request signals of the FSB. (FATAL)	0b	RWC

13.2.1.20 Offset 62 - 63h: FSB_NERR – FSB Next Error Register

Table 230. Offset 62 - 63h: FSB_NERR – FSB Next Error Register

Device: 0	**Function:** 1
Offset: 62 - 63h	**Size:** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
See Section 13.2.1.19, "Offset 60 - 61h: FSB_FERR – FSB First Error Register" on page 362 for bit definitions.				

13.2.1.21 Offset 64 - 65h: FSB_EMASK – FSB Error Mask Register

This register masks the FSB unit errors from being recognized, preventing them from being logged at the unit or global level, and no interrupt/messages are generated. These bits are sticky through reset.

Table 231. Offset 64 - 65h: FSB_EMASK – FSB Error Mask Register (Sheet 1 of 2)

Device: 0	**Function:** 1
Offset: 64 - 65h	**Size:** 16 bit
Default Value: 0009h	

Bits	Name	Description	Reset Value	Access
15:10	Reserved	Reserved	00h	
09	PARIOM	**FSB Parity error on outgoing data from I/O Mask:** This bit is sticky through reset. 0 = Enable FSB parity error from I/O detection and reporting 1 = Mask FSB parity error from I/O detection and reporting	0b	RW
08	PARMM	**FSB Parity error on outgoing data from memory Mask:** This bit is sticky through reset. 0 = Enable FSB parity error from memory detection and reporting 1 = Mask FSB parity error from memory detection and reporting	0b	RW
07	FSB_BINIT#M	**FSB BINIT# detected Mask:** This bit is sticky through reset. 0 = Enable FSB BINIT# detection and reporting 1 = Mask FSB BINIT# detection and reporting	0b	RW

Table 231. Offset 64 - 65h: FSB_EMASK – FSB Error Mask Register (Sheet 2 of 2)

Device: 0	**Function:** 1
Offset: 64 - 65h	**Size:** 16 bit
Default Value: 0009h	

Bits	Name	Description	Reset Value	Access
06	FSB_MCERR#M	**FSB MCERR# detected Mask:** This bit is sticky through reset. 0 = Enable FSB MCERR# detection and reporting 1 = Mask FSB MCERR# detection and reporting	0b	RW
05	NDLOCKM	**Non-DRAM Lock Error Mask:** This bit is sticky through reset. 0 = Enable Non-DRAM Lock Error detection and reporting 1 = Mask Non-DRAM Lock Error detection and reporting	0b	RW
04	ATOMM	**FSB Address Above TOM Mask:** This bit is sticky through reset. 0 = Enable FSB address above TOM detection and reporting 1 = Mask FSB address above TOM detection and reporting	0b	RW
03	FSBDPARM	**FSB Data Parity Error Mask:** This bit is sticky through reset. Note that FSB Data Parity Errors are masked at boot. 0 = Enable FSB Data Parity Error detection and reporting 1 = Mask FSB Data Parity Error detection and reporting	1b	RW
02	FSBAGLM	**FSB Address Strobe Glitch Detected Mask:** This bit is sticky through reset. 0 = Enable FSB address strobe glitch detection and reporting 1 = Mask FSB address strobe glitch detection and reporting	0b	RW
01	FSBDGLM	**FSB Data Strobe Glitch Detected Mask:** This bit is sticky through reset. 0 = Enable FSB data strobe glitch detection and reporting 1 = Mask FSB data strobe glitch detection and reporting	0b	RW
00	FSBRPARM	**FSB Request/Address Parity Error Detected Mask:** This bit is sticky through reset. Note that FSB Request/Address Parity Errors are masked at boot. 0 = Enable FSB request/address parity error detection and reporting 1 = Mask FSB request/address detection and reporting	1b	RW

13.2.1.22 Offset 68 - 69h: FSB_SCICMD – FSB SCI Command Register

This register enables various errors to generate an SCI NSI special cycle. When an error flag is set in the FERR or NERR registers, it generates an SCI NSI special cycle when enabled in the SCICMD register. Note that one and only one message type can be enabled.

Table 232. Offset 68 - 69h: FSB_SCICMD – FSB SCI Command Register (Sheet 1 of 2)

Device: 0	**Function:** 1
Offset: 68 - 69h	**Size:** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
15:10	Reserved	Reserved	00h	

Table 232. Offset 68 - 69h: FSB_SCICMD – FSB SCI Command Register (Sheet 2 of 2)

Device: 0 ***Function:*** 1
Offset: *68 - 69h* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
09	PARDIO	**Parity error detected on outgoing data from I/O:** Generate SCI whenever bit 9 of FSB_FERR or FSB_NERR is set. 0 = No SCI generated on parity error from I/O detection 1 = Enable SCI generation on parity error from I/O detection	0b	RW
08	PARDM	**Parity error detected on outgoing data from memory:** Generate SCI whenever bit 8 of FSB_FERR or FSB_NERR is set. 0 = No SCI generated on parity error from memory detection 1 = Enable SCI generation on parity error from memory detection	0b	RW
07	BINIT#_SCI	**FSB BINIT# detected SCI Enable:** Controls whether or not an SCI is generated when bit 7 of the FSB_FERR or FSB_NERR register is set. 0 = No SCI generated on FSB BINIT# detection 1 = Enable SCI generation on FSB BINIT# detection	0b	RW
06	MCERR#_SCI	**FSB MCERR# detected SCI Enable:** Controls whether or not an SCI is generated when bit 6 of the FSB_FERR or FSB_NERR register is set. 0 = No SCI generated on FSB MCERR# detection 1 = Enable SCI generation on FSB MCERR# detection	0b	RW
05	NDLOCK_SCI	**Non-DRAM Lock Error SCI Enable:** Controls whether or not an SCI is generated when bit 5 of the FSB_FERR or FSB_NERR register is set. 0 = No SCI generated on Non-DRAM Lock Error detection 1 = Enable SCI generation on Non-DRAM Lock Error detection	0b	RW
04	ATOM_SCI	**FSB Address Above TOM SCI Enable:** Controls whether or not an SCI is generated when bit 4 of the FSB_FERR or FSB_NERR register is set. 0 = No SCI generated on FSB address above TOM detection 1 = Enable SCI generation on FSB address above TOM detection	0b	RW
03	FSBDPAR_SCI	**FSB Data Parity Error SCI Enable:** Controls whether or not an SCI is generated when bit 3 of the FSB_FERR or FSB_NERRR register is set. 0 = No SCI generated on FSB Data Parity Error detection 1 = Enable SCI generation on FSB Data Parity Error detection	0b	RW
02	FSBAGL_SCI	**FSB Address Strobe Glitch Detected SCI Enable:** Controls whether or not an SCI is generated when bit 2 of the FSB_FERR or FSB_NERR register is set. 0 = No SCI generated on FSB address strobe glitch detection 1 = Enable SCI generation on FSB address strobe glitch detection	0b	RW
01	FSBDGL_SCI	**FSB Data Strobe Glitch Detected SCI Enable:** Controls whether or not an SCI is generated when bit 1 of the FSB_FERR or FSB_NERR register is set. 0 = No SCI generated on FSB data strobe glitch detection 1 = Enable SCI generation on FSB data strobe glitch detection	0b	RW
00	FSBRPAR_SCI	**FSB Request/Address Parity Error Detected SCI Enable:** Controls whether or not an SCI is generated when bit 0 of the FSB_FERR or FSB_NERR register is set. 0 = No SCI generated on FSB request/address parity error detection 1 = Enable SCI generation on FSB request/address detection	0b	RW

13.2.1.23 Offset 6A - 6Bh: FSB_SMICMD – FSB SMI Command Register

This register enables various errors to generate an SMI NSI special cycle. When an error flag is set in the FSB_FERR or FSB_NERR register, it generates an SMI NSI special cycle when enabled in the SMICMD register. Note that one and only one message type can be enabled.

Table 233. Offset 6A - 6Bh: FSB_SMICMD – FSB SMI Command Register (Sheet 1 of 2)

Device: 0 ***Function:*** 1
Offset: *6A - 6Bh* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:10	Reserved	Reserved	00h	
09	POUTIO	**Parity error detected on outgoing data from I/O:** Generate SMI whenever bit 9 of FSB_FERR or FSB_NERR is set. 0 = No SMI generated on parity error from I/O detection 1 = Enable SMI generation on parity error from I/O detection	0b	RW
08	POUTM	**Parity error detected on outgoing data from memory:** Generate SMI whenever bit 8 of FSB_FERR or FSB_NERR is set. 0 = No SMI generated on parity error from memory detection 1 = Enable SMI generation on parity error from memory detection	0b	RW
07	BINIT#_SMI	**FSB BINIT# detected SMI Enable:** Controls whether or not an SMI is generated when bit 7 of the FSB_FERR or FSB_NERR register is set. 0 = No SMI generated on FSB BINIT# detection 1 = Enable SMI generation on FSB BINIT# detection	0b	RW
06	MCERR#_SMI	**FSB MCERR# detected SMI Enable:** Controls whether or not an SMI is generated when bit 6 of the FSB_FERR or FSB_NERR register is set. 0 = No SMI generated on FSB MCERR# detection 1 = Enable SMI generation on FSB MCERR# detection	0b	RW
05	NDLOCK_SMI	**Non-DRAM Lock Error SMI Enable:** Controls whether or not an SMI is generated when bit 5 of the FSB_FERR or FSB_NERR register is set. 0 = No SMI generated on Non-DRAM Lock Error detection 1 = Enable SMI generation on Non-DRAM Lock Error detection	0b	RW
04	ATOM_SMI	**FSB Address Above TOM SMI Enable:** Controls whether or not an SMI is generated when bit 4 of the FSB_FERR or FSB_NERR register is set. 0 = No SMI generated on FSB address above TOM detection 1 = Enable SMI generation on FSB address above TOM detection	0b	RW
03	FSBDPAR_SMI	**FSB Data Parity Error SMI Enable:** Controls whether or not an SMI is generated when bit 3 of the FSB_FERR or FSB_NERRR register is set. 0 = No SMI generated on FSB Data Parity Error detection 1 = Enable SMI generation on FSB Data Parity Error detection	0b	RW

Table 233. Offset 6A - 6Bh: FSB_SMICMD – FSB SMI Command Register (Sheet 2 of 2)

Device: 0	***Function:*** 1
Offset: *6A - 6Bh*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
02	FSBAGL_SMI	**FSB Address Strobe Glitch Detected SMI Enable:** Controls whether or not an SMI is generated when bit 2 of the FSB_FERR or FSB_NERR register is set. 0 = No SMI generated on FSB address strobe glitch detection 1 = Enable SMI generation on FSB address strobe glitch detection	0b	RW
01	FSBDGL_SMI	**FSB Data Strobe Glitch Detected SMI Enable:** Controls whether or not an SMI is generated when bit 1 of the FSB_FERR or FSB_NERR register is set. 0 = No SMI generated on FSB data strobe glitch detection 1 = Enable SMI generation on FSB data strobe glitch detection	0b	RW
00	FSBRPAR_SMI	**FSB Request/Address Parity Error Detected SMI Enable:** Controls whether or not an SMI is generated when bit 0 of the FSB_FERR or FSB_NERR register is set. 0 = No SMI generated on FSB request/address parity error detection 1 = Enable SMI generation on FSB request/address detection	0b	RW

13.2.1.24 Offset 6C - 6Dh: FSB_SERRCMD – FSB SERR Command Register

This register enables various errors to generate an SERR NSI special cycle. When an error flag is set in the FSB_FERR or FSB_NERR register, it generates an SERR NSI special cycle when enabled in the SERRCMD register. Note that one and only one message type can be enabled.

Table 234. Offset 6C - 6Dh: FSB_SERRCMD – FSB SERR Command Register (Sheet 1 of 2)

Device: 0	***Function:*** 1
Offset: *6C - 6Dh*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:10	Reserved	Reserved	00h	
09	PEDIO	**Parity error detected on outgoing data from I/O:** Generate SERR whenever bit 9 of FSB_FERR or FSB_NERR is set. 0 = No SERR generated on parity error from I/O detection 1 = Enable SERR generation on parity error from I/O detection	0b	RW
08	PEDM	**Parity error detected on outgoing data from memory:** Generate SERR whenever bit 8 of FSB_FERR or FSB_NERR is set. 0 = No SERR generated on parity error from memory detection 1 = Enable SERR generation on parity error from memory detection	0b	RW
07	BINIT#_SERR	**FSB BINIT# detected SERR Enable:** Controls whether or not an SERR is generated when bit 7 of the FSB_FERR or FSB_NERR register is set. 0 = No SERR generated on FSB BINIT# detection 1 = Enable SERR generation on FSB BINIT# detection	0b	RW

Table 234. Offset 6C - 6Dh: FSB_SERRCMD – FSB SERR Command Register (Sheet 2 of 2)

Device: 0 **Function:** 1
Offset: 6C - 6Dh **Size:** 16 bit
Default Value: 0000h

Bits	Name	Description	Reset Value	Access
06	MCERR#_SERR	**FSB MCERR# detected SERR Enable:** Controls whether or not an SERR is generated when bit 6 of the FSB_FERR or FSB_NERR register is set. 0 = No SERR generated on FSB MCERR# detection 1 = Enable SERR generation on FSB MCERR# detection	0b	RW
05	NDLOCK_SERR	**Non-DRAM Lock Error SERR Enable:** Controls whether or not an SERR is generated when bit 5 of the FSB_FERR or FSB_NERR register is set. 0 = No SERR generated on Non-DRAM Lock Error detection 1 = Enable SERR generation on Non-DRAM Lock Error detection	0b	RW
04	ATOM_SERR	**FSB Address Above TOM SERR Enable:** Controls whether or not an SERR is generated when bit 4 of the FSB_FERR or FSB_NERR register is set. 0 = No SERR generated on FSB address above TOM detection 1 = Enable SERR generation on FSB address above TOM detection	0b	RW
03	FSBDPAR_SERR	**FSB Data Parity Error SERR Enable:** Controls whether or not an SERR is generated when bit 3 of the FSB_FERR or FSB_NERRR register is set. 0 = No SERR generated on FSB Data Parity Error detection 1 = Enable SERR generation on FSB Data Parity Error detection	0b	RW
02	FSBAGL_SERR	**FSB Address Strobe Glitch Detected SERR Enable:** Controls whether or not an SERR is generated when bit 2 of the FSB_FERR or FSB_NERR register is set. 0 = No SERR generated on FSB address strobe glitch detection 1 = Enable SERR generation on FSB address strobe glitch detection	0b	RW
01	FSBDGL_SERR	**FSB Data Strobe Glitch Detected SERR Enable:** Controls whether or not an SERR is generated when bit 1 of the FSB_FERR or FSB_NERR register is set. 0 = No SERR generated on FSB data strobe glitch detection 1 = Enable SERR generation on FSB data strobe glitch detection	0b	RW
00	FSBRPAR_SERR	**FSB Request/Address Parity Error Detected SERR Enable:** Controls whether or not an SERR is generated when bit 0 of the FSB_FERR or FSB_NERR register is set. 0 = No SERR generated on FSB request/address parity error detection 1 = Enable SERR generation on FSB request/address detection	0b	RW

13.2.1.25 Offset 6E - 6Fh: FSB_MCERRCMD – FSB MCERR Command Register

This register enables various errors to generate the MCERR signal on the FSB. When an error flag is set in the FSB_FERR or FSB_NERR register, it generates a MCERR# when enabled in the MCERRCMD.

Table 235. Offset 6E - 6Fh: FSB_MCERRCMD – FSB MCERR Command Register (Sheet 1 of 2)

Device: 0	***Function:*** 1
Offset: *6E - 6Fh*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:10	Reserved	Reserved	00h	
09	PERRIO	**Parity error detected on outgoing data from I/O:** Generate MCERR whenever bit 9 of FSB_FERR or FSB_NERR is set. 0 = No MCERR# generated on parity error from I/O detection 1 = Enable MCERR# generation on parity error from I/O detection	0b	RW
08	PERRM	**Parity error detected on outgoing data from memory:** Generate MCERR whenever bit 8 of FSB_FERR or FSB_NERR is set. 0 = No MCERR# generated on parity error from memory detection 1 = Enable MCERR# generation on parity error from memory detection	0b	RW
07	BINIT#_MCERR#	**FSB BINIT# detected MCERR# Enable:** Controls whether or not an MCERR# is generated when bit 7 of the FSB_FERR or FSB_NERR register is set. 0 = No MCERR# generated on FSB BINIT# detection 1 = Enable MCERR# generation on FSB BINIT# detection	0b	RW
06	FSBMCERR#	**FSB MCERR# detected MCERR# Enable:** Controls whether or not an MCERR# is generated when bit 6 of the FSB_FERR or FSB_NERR register is set. 0 = No MCERR# generated on FSB MCERR# detection 1 = Enable MCERR# generation on FSB MCERR# detection	0b	RW
05	NDLOCK_MCERR#	**Non-DRAM Lock Error MCERR# Enable:** Controls whether or not an MCERR# is generated when bit 5 of the FSB_FERR or FSB_NERR register is set. 0 = No MCERR# generated on Non-DRAM Lock Error detection 1 = Enable MCERR# generation on Non-DRAM Lock Error detection	0b	RW
04	ATOM_MCERR#	**FSB Address Above TOM MCERR# Enable:** Controls whether or not an MCERR# is generated when bit 4 of the FSB_FERR or FSB_NERR register is set. 0 = No MCERR# generated on FSB address above TOM detection 1 = Enable MCERR# generation on FSB address above TOM detection	0b	RW
03	FSBDPAR_MCERR#	**FSB Data Parity Error MCERR# Enable:** Controls whether or not an MCERR# is generated when bit 3 of the FSB_FERR or FSB_NERRR register is set. 0 = No MCERR# generated on FSB Data Parity Error detection 1 = Enable MCERR# generation on FSB Data Parity Error detection	0b	RW

Table 235. Offset 6E - 6Fh: FSB_MCERRCMD – FSB MCERR Command Register (Sheet 2 of 2)

Device: 0	**Function:** 1
Offset: 6E - 6Fh	**Size:** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
02	FSBAGL_MCERR#	**FSB Address Strobe Glitch Detected MCERR# Enable:** Controls whether or not an MCERR# is generated when bit 2 of the FSB_FERR or FSB_NERR register is set. 0 = No MCERR# generated on FSB address strobe glitch detection 1 = Enable MCERR# generation on FSB address strobe glitch detection	0b	RW
01	FSBDGL_MCERR#	**FSB Data Strobe Glitch Detected MCERR# Enable:** Controls whether or not an MCERR# is generated when bit 1 of the FSB_FERR or FSB_NERR register is set. 0 = No MCERR# generated on FSB data strobe glitch detection 1 = Enable MCERR# generation on FSB data strobe glitch detection	0b	RW
00	FSBRPAR_MCERR#	**FSB Request/Address Parity Error Detected MCERR# Enable:** Controls whether or not an MCERR# is generated when bit 0 of the FSB_FERR or FSB_NERR register is set. 0 = No MCERR# generated on FSB request/address parity error detection 1 = Enable MCERR# generation on FSB request/address detection	0b	RW

13.2.1.26 Offset 70h: BUF_FERR – Memory Buffer First Error Register

Signals errors occurring in the memory system coherent Posted Memory Write Buffer (PMWB). This register is sticky through reset.

13.2.1.27 Offset 72h: BUF_NERR – Memory Buffer Next Error Register

Table 236. Offset 72h: BUF_NERR – Memory Buffer Next Error Register

Device: 0	**Function:** 1
Offset: 72h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
See Section 13.2.1.26, "Offset 70h: BUF_FERR – Memory Buffer First Error Register" on page 371 for bit definitions.				

13.2.1.28 Offset 74h: BUF_EMASK – Memory Buffer Error Mask Register

This register masks the unit errors from being recognized. Because they are not recognized, they are not logged at the unit or global level and no interrupt/messages are generated. This register is sticky through reset.

Table 237. Offset 74h: BUF_EMASK – Memory Buffer Error Mask Register

Device: 0	***Function:*** 1
Offset: *74h*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	00h	
03	BUF_EMASK03	Mask Error bit 03: 0 = Disable mask. 1 = Enable mask.	0b	RW
02	BUF_EMASK02	Mask Error bit 02: 0 = Disable mask. 1 = Enable mask.	0b	RW
01	BUF_EMASK01	Mask Error bit 01: 0 = Disable mask. 1 = Enable mask.	0b	RW
00	BUF_EMASK00	Mask Error bit 00: 0 = Disable mask. 1 = Enable mask.	0b	RW

13.2.1.29 Offset 78h: BUF_SCICMD – Memory Buffer SCI Command Register

This register enables various errors to generate an SCI NSI special cycle. When an error flag is set in the FERR or NERR registers, it generates an SCI NSI special cycle when enabled in the SCICMD registers. Note that one and only one message type can be enabled.

Table 238. Offset 78h: BUF_SCICMD – Memory Buffer SCI Command Register (Sheet 1 of 2)

Device: 0	***Function:*** 1
Offset: *78h*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	0h	
03	DPMWB_SCI	**Internal DRAM Interface to PMWB Parity Error SCI Enable:** Generate SCI when parity error detected for DRAM Interface to PMWB when this bit is set. 0 = Disable 1 = Enable	0b	RW

Table 238. Offset 78h: BUF_SCICMD – Memory Buffer SCI Command Register (Sheet 2 of 2)

Device: 0	***Function:*** 1
Offset: *78h*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
02	IOPMWB_SCI	**Internal System Bus or I/O to PMWB Parity Error SCI Enable:** Generate SCI when parity error detected for internal System Bus or I/O to PMWB when this bit is set. 0 = Disable 1 = Enable	0b	RW
01	PMWBSYS_SCI	**Internal PMWB to System Bus Parity Error SCI Enable:** Generate SCI when parity error detected for PMWB to System Bus when this bit is set. 0 = Disable 1 = Enable	0b	RW
00	PMWBD_SCI	**Internal PMWB to DRAM I/F Parity Error SCI Enable:** Generate SCI when parity error detected for PMWB to DRAM I/F when this bit is set. 0 = Disable 1 = Enable	0b	RW

13.2.1.30 Offset 7Ah: BUF_SMICMD – Memory Buffer SMI Command Register

This register enables various errors to generate an SMI NSI special cycle. When an error flag is set in the FERR or NERR registers, it generates an SMI NSI special cycle when enabled in the SMICMD register. Note that one and only one message type can be enabled.

Table 239. Offset 7Ah: BUF_SMICMD – Memory Buffer SMI Command Register (Sheet 1 of 2)

Device: 0	***Function:*** 1
Offset: *7Ah*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	0h	
03	DPMWB_SMI	**Internal DRAM Interface to PMWB Parity Error SMI Enable:** Generate SMI when parity error detected for DRAM interface to PMWB when this bit is set. 0 = Disable 1 = Enable	0b	RW
02	IOPMWB_SMI	**Internal System Bus or I/O to PMWB Parity Error SMI Enable:** Generate SMI when parity error detected for internal System Bus or I/O to PMWB when this bit is set. 0 = Disable 1 = Enable	0b	RW
01	PMWBSYS_SMI	**Internal PMWB to System Bus Parity Error SMI Enable:** Generate SMI when parity error detected for PMWB to System Bus when this bit is set. 0 = Disable 1 = Enable	0b	RW

Table 239. Offset 7Ah: BUF_SMICMD – Memory Buffer SMI Command Register (Sheet 2 of 2)

Device: 0	**Function:** 1
Offset: 7Ah	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
00	PMWBD_SMI	**Internal PMWB to DRAM I/F Parity Error SMI Enable:** Generate SMI when parity error detected for PMWB to DRAM I/F when this bit is set. 0 = Disable 1 = Enable	0b	RW

13.2.1.31 Offset 7Ch: BUF_SERRCMD – Memory Buffer SERR Command Register

This register enables various errors to generate an SERR NSI special cycle. When an error flag is set in the FERR or NERR registers, it generates an SERR NSI special cycle when enabled in the SERRCMD register. Note that one and only one message type can be enabled.

Table 240. Offset 7Ch: BUF_SERRCMD – Memory Buffer SERR Command Register

Device: 0	**Function:** 1
Offset: 7Ch	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	0h	
03	DPMWB_SERR	**Internal DRAM I/F to PMWB Parity Error SERR Enable:** Generate SERR when parity error detected for DRAM I/F to PMWB when this bit is set. 0 = Disable 1 = Enable	0b	RW
02	IOPMWB_SERR	**Internal System Bus or I/O to PMWB Parity Error SERR Enable:** Generate SERR when parity error detected on write port 0 when this bit is set. 0 = Disable 1 = Enable	0b	RW
01	PMWBSYS_SERR	**Internal PMWB to System Bus Parity Error SERR Enable:** Generate SERR when parity error detected for PMWB to System Bus when this bit is set. 0 = Disable 1 = Enable	0b	RW
00	PMWBD_SERR	**Internal PMWB to DRAM I/F Parity Error SERR Enable:** Generate SERR when parity error detected for PMWB to DRAM I/F when this bit is set. 0 = Disable 1 = Enable	0b	RW

13.2.1.32 Offset 7Eh: BUF_MCERRCMD – Memory Buffer MCERR Command Register

This register enables various errors to generate a MCERR signal on the FSB. When an error flag is set in the FERR or NERR registers, it generates a MCERR when enabled in the MCERRCMD register.

Table 241. Offset 7Eh: BUF_MCERRCMD – Memory Buffer MCERR Command Register

Device: 0	***Function:*** 1
Offset: *7Eh*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	0h	
03	DPMWB_MCERR	**Internal DRAM I/F to PMWB Parity Error MCERR Enable:** Generate MCERR when parity error detected for DRAM I/F to PMWB when this bit is set. 0 = Disable 1 = Enable	0b	RW
02	IOPMWB_MCERR	**Internal System Bus or I/O to PMWB Parity Error MCERR Enable:** Generate MCERR when parity error detected on write port 0 when this bit is set. 0 = Disable 1 = Enable	0b	RW
01	PMWBSYS_MCERR	**Internal PMWB to System Bus Parity Error MCERR Enable:** Generate MCERR when parity error detected for PMWB to System Bus when this bit is set. 0 = Disable 1 = Enable	0b	RW
00	PMWBD_MCERR	**Internal PMWB to DRAM I/F Parity Error MCERR Enable:** Generate MCERR when parity error detected for PMWB to DRAM I/F when this bit is set. 0 = Disable 1 = Enable	0b	RW

13.2.1.33 Offset 80 - 81h: DRAM_FERR – DRAM First Error Register

This register stores the first error related to the DRAM Controller. Only one error bit is set in this register. Any future errors (NEXT errors) are set in the DRAM_NERR register. No further error bits in the DRAM_FERR register are set until the existing error bit is cleared. These bits are sticky through reset. Software clears these bits by writing a 1 to the bit location.

Note: If multiple errors are reported in the same clock as the first error, all errors are latched.

Table 242. Offset 80 - 81h: DRAM_FERR – DRAM First Error Register

Device: 0 **Function:** 1
Offset: 80 - 81h **Size:** 16 bit
Default Value: 0000h

Bits	Name	Description	Reset Value	Access
15:08	Reserved	**Reserved**	00h	
07	MTCA	**Memory Test Complete:** Not an error condition. This bit is set by hardware to signal BIOS that hardware testing of the channel is complete. This bit is sticky through reset. 0 = System software clears this bit by writing a 1 to the location. 1 = Hardware-based test of DRAM is complete. (NON-FATAL)	0b	RWC
06	UERRA	**Uncorrectable Error on Write:** (Uncorrectable) This bit is set on a detected error regardless of ECC mode, even if ECC is disabled. However if the error was injected via ECCDIAG register, this bit is not set. This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No poisoned write to DRAM detected. 1 = Poisoned write to DRAM detected. (NON-FATAL)	0b	RWC
05	DEDA	**DED Retry Initiated:** (Uncorrectable) This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No DED Retry Initiated 1 = DED Retry Initiated. This can be set for a normal demand data read, or for a scrub that is retried. (NON-FATAL)	0b	RWC
04	Reserved	Reserved	0b	
03	ETDA	**Error Threshold Detect:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. This bit can be set by either a SEC or DED event, if the corresponding DIMM error counter is set. 0 = No Error Threshold detected 1 = Error Threshold detected. (NON-FATAL)	0b	RWC
02	USDEA	**Uncorrectable Scrubber Data Error:** This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No Scrubber Error Detected 1 = Scrubber Error Detected. (NON-FATAL)	0b	RWC
01	URMEA	**Uncorrectable Read Memory Error:** (Uncorrectable) Applies to non-scrub (normal demand fetch) reads. This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No Uncorrectable Non-Scrub Demand Read Memory Error 1 = Uncorrectable Non-Scrub Demand Read Memory Error. (NON-FATAL)	0b	RWC
00	CRMEA	**Correctable read memory Error:** (Correctable) SECs (Single Bit Error Correction) detected by normal demand requests or scrub/demand fetch (normal read to memory). This bit is sticky through reset. System software clears this bit by writing a 1 to the location. 0 = No Correctable Read Memory Error. 1 = Correctable Read Memory Error. (NON-FATAL)	0b	RWC

13.2.1.34 Offset 82 - 83h: DRAM_NERR – DRAM Next Error Register

Table 243. Offset 82 - 83h: DRAM_NERR – DRAM Next Error Register

Device: 0			**Function:** 1	
Offset: 82 - 83h			**Size:** 16 bit	
Default Value: 0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.2.1.33, "Offset 80 - 81h: DRAM_FERR – DRAM First Error Register" on page 375 for bit definitions.				

13.2.1.35 Offset 84h: DRAM_EMASK – DRAM Error Mask Register

This register masks the DRAM Controller errors and events from being recognized, preventing them from being logged at the unit or global level, and no interrupt/ messages are generated. These bits are sticky through reset.

Table 244. Offset 84h: DRAM_EMASK – DRAM Error Mask Register (Sheet 1 of 2)

Device: 0			**Function:** 1	
Offset: 84h			**Size:** 8 bit	
Default Value: 00h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
07	MTC_MASK	**Memory Test Complete Mask:** This bit is sticky through reset. 0 = Allow Memory Test Complete logging and signaling. 1 = Mask Memory Test Complete logging and signaling.	0b	RW
06	UERR_MASK	**Uncorrectable Error Detected on Write to DRAM Mask:** This bit is sticky through reset. 0 = Allow Poisoned Write to DRAM detection and signaling. 1 = Mask Poisoned Write to DRAM detection and signaling.	0b	RW
05	DED_MASK	**DED Retry Initiated Mask:** This bit is sticky through reset. 0 = Allow DED Retry Initiated detection and signaling. 1 = Mask DED Retry Initiated detection and signaling.	0b	RW
04	RSVD	**Reserved**	0b	
03	ETD_MASK	**Error Threshold Detect Mask:** This bit is sticky through reset. 0 = Allow Error Threshold detection and signaling. 1 = Mask Error Threshold detection and signaling.	0b	RW

Table 244. Offset 84h: DRAM_EMASK – DRAM Error Mask Register (Sheet 2 of 2)

Device: 0	**Function:** 1
Offset: 84h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
02	SDE_MASK	**Scrubber Data Error Mask:** This bit is sticky through reset. 0 = Allow Scrubber Data Error detection and signaling. 1 = Mask Scrubber Data Error detection and signaling.	0b	RW
01	URME_MASk	**Uncorrectable Read Memory Error Mask:** This bit is sticky through reset. 0 = Allow Uncorrectable Memory Read Error detection and signaling. 1 = Mask Uncorrectable Memory Read Error detection and signaling.	0b	RW
00	CRME_MASK	**Correctable Read Memory Error Mask:** This bit is sticky through reset. 0 = Allow Correctable Memory Read Error detection and signaling. 1 = Mask Correctable Memory Read Error detection and signaling.	0b	RW

13.2.1.36 Offset 88h: DRAM_SCICMD – DRAM SCI Command Register

This register enables various errors to generate an SCI NSI special cycle. When an error flag is set in the FERR or NERR registers, it generates an SERR, SMI, or SCI NSI special cycle when enabled in the ERRCMD, SMICMD, or SCICMD registers, respectively. Note that one and only one message type can be enabled.

Table 245. Offset 88h: DRAM_SCICMD – DRAM SCI Command Register (Sheet 1 of 2)

Device: 0	**Function:** 1
Offset: 88h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07	MTC_SCI	**Memory Test Complete SCI Enable:** Generate SCI when Bit 7 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
06	PWDRAM_SCI	**Poisoned Write to DRAM SCI Enable:** Generate SCI when Bit 6 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
05	DED_SCI	**DED Retry Initiated SCI Enable:** Generate SCI when Bit 5 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
04	RSVD	**Reserved**	0b	
03	EDT_SCI	**Error Threshold Detect SCI Enable:** Generate SCI when Bit 3 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW

Table 245. Offset 88h: DRAM_SCICMD – DRAM SCI Command Register (Sheet 2 of 2)

Device: 0	**Function:** 1
Offset: 88h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
02	SDE_SCI	**Scrubber Data Error SCI Enable:** Generate SCI when Bit 2 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
01	URME_SCI	**Uncorrectable Read Memory Error SCI Enable:** Generate SCI when Bit 1 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
00	CRME_SCI	**Correctable Read Memory Error SCI Enable:** Generate SCI when Bit 0 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW

13.2.1.37 Offset 8AH: DRAM_SMICMD – DRAM SMI Command Register

This register enables various errors to generate an SMI NSI special cycle. When an error flag is set in the FERR or NERR registers, it generates an SERR, SMI, or SCI NSI special cycle when enabled in the ERRCMD, SMICMD, or SCICMD registers, respectively. Note that one and only one message type can be enabled.

Table 246. Offset 8AH: DRAM_SMICMD – DRAM SMI Command Register (Sheet 1 of 2)

Device: 0	**Function:** 1
Offset: 8Ah	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07	MTC_SMI	**Memory Test Complete SMI Enable:** Generate SMI when Bit 7 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
06	PWDRAM_SMI	**Poisoned Write to DRAM SMI Enable:** Generate SMI when Bit 6 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
05	DED_SMI	**DED Retry Initiated SMI Enable:** Generate SMI when Bit 5 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
04	RSVD_	**Reserved**	0b	
03	ETD_SMI	**Error Threshold Detect SMI Enable:** Generate SMI when Bit 3 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW

Table 246. Offset 8AH: DRAM_SMICMD – DRAM SMI Command Register (Sheet 2 of 2)

Device: 0	**Function:** 1
Offset: 8Ah	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
02	SDE_SMI	**Scrubber Data Error SMI Enable:** Generate SMI when Bit 2 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
01	URME_SMI	**Uncorrectable Read Memory Error SMI Enable:** Generate SMI when Bit 1 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
00	CRME_SMI	**Correctable Read Memory Error SMI Enable:** Generate SMI when Bit 0 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW

13.2.1.38 Offset 8Ch: DRAM_SERRCMD – DRAM SERR Command Register

This register enables various errors to generate an SERR NSI special cycle. When an error flag is set in the FERR or NERR registers, it generates an SERR, SMI, or SCI NSI special cycle when enabled in the SERRCMD, SMICMD, or SCICMD registers, respectively. Note that one and only one message type can be enabled.

Table 247. Offset 8Ch: DRAM_SERRCMD – DRAM SERR Command Register (Sheet 1 of 2)

Device: 0	**Function:** 1
Offset: *8Ch*	**Size:** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07	MTC_SERR	**Memory Test Complete SERR Enable:** Generate SERR when Bit7 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
06	PWDRAM_SERR	**Poisoned Write to DRAM SERR Enable:** Generate SERR when Bit 6 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
05	DED_SERR	**DED Retry Initiated SERR Enable:** Generate SERR when Bit 5 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
04	RSVD	1 = **Reserved**	0b	
03	ETD_SERR	**Error Threshold Detect SERR Enable:** Generate SERR when Bit 3 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW

Table 247. Offset 8Ch: DRAM_SERRCMD – DRAM SERR Command Register (Sheet 2 of 2)

Device: 0	**Function:** 1
Offset: 8Ch	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
02	SDE_SERR	**Scrubber Data Error SERR Enable:** Generate SERR when Bit2 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
01	URME_SERR	**Uncorrectable Read Memory Error SERR Enable:** Generate SERR when Bit 1 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
00	CRME_SERR	**Correctable Read Memory Error SERR Enable:** Generate SERR when Bit 0 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW

13.2.1.39 Offset 8Eh: DRAM_MCERRCMD – DRAM MCERR Command Register

This register enables various errors to generate a MCERR# signal on the FSB. When an error flag is set in the FERR or NERR registers, it generates a MCERR#, SMI, or SCI NSI special cycle when enabled in the MCERRCMD register.

Table 248. Offset 8Eh: DRAM_MCERRCMD – DRAM MCERR Command Register (Sheet 1 of 2)

Device: 0	**Function:** 1
Offset: 8Eh	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07	MTC_MCERR#	**Memory Test Complete MCERR# Enable:** Generate MCERR# when Bit 7 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
06	PWDRAM_MCERR#	**Poisoned Write to DRAM MCERR# Enable:** Generate MCERR# when Bit 6 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
05	DED_MCERR#	**DED Retry Initiated MCERR# Enable:** Generate MCERR# when Bit 5 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
04	RSVD	**Reserved**	0b	
03	ETD_MCERR#	**Error Threshold Detect MCERR# Enable:** Generate MCERR# when Bit 3 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW

Table 248. Offset 8Eh: DRAM_MCERRCMD – DRAM MCERR Command Register (Sheet 2 of 2)

Device: 0	**Function:** 1
Offset: 8Eh	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
02	SDE_MCERR#	**Scrubber Data Error MCERR# Enable:** Generate MCERR# when Bit 2 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
01	URME_MCERR#	**Uncorrectable Read Memory ErrorMCERR# Enable:** Generate MCERR# when Bit 1 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW
00	CRME_MCERR#	**Correctable Read Memory Error MCERR# Enable:** Generate MCERR# when Bit 0 of DRAM_FERR or DRAM_NERR is set. 0 = Disable 1 = Enable	0b	RW

13.2.1.40 Offset 98 - 99h: THRESH_SEC0 – DIMM0 SEC Threshold Register

Threshold compare value for SEC errors. An Error Threshold Detect is signaled if the logical DIMM0 SEC counter (Section 13.2.1.48) exceeds the value programmed into this register. The bits in this register are sticky through reset.

Table 249. Offset 98 - 99h: THRESH_SEC0 – DIMM0 SEC Threshold Register

Device: 0	**Function:** 1
Offset: 98 - 99h	**Size:** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
15:00	THRESH_SEC0	Threshold compare value for logical DIMM0 SEC errors.	0000h	RW

13.2.1.41 Offset 9A - 9Bh: THRESH_SEC1 – DIMM1 SEC Threshold Register

Threshold compare value for SEC errors. An Error Threshold Detect is signaled if the logical DIMM1 SEC counter (Section 13.2.1.50) exceeds the value programmed into this register. The bits in this register are sticky through reset.

Table 250. Offset 9A - 9Bh: THRESH_SEC1 – DIMM1 SEC Threshold Register

Device: 0	**Function:** 1
Offset: 9A - 9Bh	**Size:** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
15:00	THRESH_SEC1	Threshold compare value for logical DIMM1 SEC errors.	0000h	RW

13.2.1.42 Offset 9C - 9Dh: THRESH_SEC2 – DIMM2 SEC Threshold Register

Threshold compare value for SEC errors. An Error Threshold Detect is signaled if the logical DIMM2 SEC counter (Section 13.2.1.52) exceeds the value programmed into this register. The bits in this register are sticky through reset.

Table 251. Offset 9C - 9Dh: THRESH_SEC2 – DIMM2 SEC Threshold Register

Device: 0 ***Function:*** 1
Offset: *9C - 9Dh* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:00	THRESH_SEC2	Threshold compare value for logical DIMM2 SEC errors.	0000h	RW

13.2.1.43 Offset 9E - 9Fh: THRESH_SEC3 – DIMM3 SEC Threshold Register

Threshold compare value for SEC errors. An Error Threshold Detect is signaled if the logical DIMM3 SEC counter (Section 13.2.1.54) exceeds the value programmed into this register. The bits in this register are sticky through reset.

Table 252. Offset 9E - 9Fh: THRESH_SEC3 – DIMM3 SEC Threshold Register

Device: 0 ***Function:*** 1
Offset: *9E - 9Fh* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:00	THRESH_SEC3	Threshold compare value for logical DIMM3 SEC errors.	0000h	RW

13.2.1.44 Offset A0 - A3h: DRAM_SECF_ADD – DRAM First Single Bit Error Correct Address Register

Captures the address of the SEC error occurring in the memory system (including scrubs). The value in this register is only valid if the Correctable Read Memory Error bit in the DRAM_FERR register has been set. The bits in this register are sticky through reset. This register saves the first address of a correctable read data error on a read, including scrubs.

Table 253. Offset A0 - A3h: DRAM_SECF_ADD – DRAM First Single Bit Error Correct Address Register

Device: 0 ***Function:*** 1
Offset: *A0 - A3* ***Size:*** *32 bit*
Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31	Reserved	Reserved	0b	
30:02	FSEFADD	**First Correctable Error Address:** This field contains address bits 34:06 for the first correctable error. This field is set by hardware, and represents a physical address. This field can only be reset by a PWRGD reset.	000_0000h	RO
01:00	Reserved	Reserved	0b	

13.2.1.45 Offset A4 - A7h: DRAM_DED_ADD – DRAM Double Bit Error Address Register

Captures the address of the first DED (uncorrectable non-scrub engine) error occurring in the memory system. If DED Retry is enabled, this register captures the address of the first failed retry for a non-scrub read. The value in this register is only valid if the Uncorrectable Read Memory Error bit in either the DRAM_FERR or DRAM_NERR register has been set. The bits in this register are sticky through reset.

Table 254. Offset A4 - A7h: DRAM_DED_ADD – DRAM Double Bit Error Address Register

Device: 0 ***Function:*** 1
Offset: *A4 - A7h* ***Size:*** *32 bit*
Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31	Reserved	Reserved	0b	
30:02	FUERRAD	**First Uncorrectable Error Address:** This field contains address bits 34:06 for the first uncorrectable error. When DED Retry is enabled, the address of the first failed retry is captured. This field is set by hardware, and represents a physical address. This field is only reset by a PWRGD reset.	000_0000h	RO
01:00	Reserved	Reserved	0b	

13.2.1.46 Offset A8 - ABh: DRAM_SCRB_ADD – DRAM Scrub Error Address Register

Captures the address of the first uncorrectable error encountered by the scrub engine for a periodic memory scrub. The bits in this register are sticky through reset.

Table 255. Offset A8 - ABh: DRAM_SCRB_ADD – DRAM Scrub Error Address Register

Device: 0 ***Function:*** 1
Offset: *A8 - ABh* ***Size:*** *32 bit*
Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31	Reserved	Reserved	0b	
30:02	SEADD	**Scrub Error Address:** This field contains address bits 34:06 for the first uncorrectable error encountered by the periodic memory scrubber. This field is set by hardware, and represents a physical address. This field is only reset by a PWRGD reset.	000_0000h	RO
01:00	Reserved	Reserved	0b	

13.2.1.47 Offset AC - AFh: DRAM_RETR_ADD – DRAM DED Retry Address Register

Captures the address of the first DED retry occurring in the memory system. This retry address can be for a normal demand data read, or for a scrub that is retried. This register is sticky through reset.

Table 256. Offset AC - AFh: DRAM_RETR_ADD – DRAM DED Retry Address Register

Device: 0	**Function:** 1
Offset: AC - AFh	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
31	Reserved	Reserved	0b	
30:02	RETRADD	**DED Retry Address:** This field contains address bits 34:6 for the first DED Retry occurring in the memory system. This field is set by hardware, and represents a physical address. This field can only be reset by a PWRGD reset.	000_0000h	RO
01:00	Reserved	Reserved	0b	

13.2.1.48 Offset B0 - B1h: DRAM_SEC_D0A – DRAM SEC Logical DIMM 0 Counter Register

Counter for SEC errors occurring for logical DIMM0 in the memory system. The DIMM counters for SEC & DED errors are implemented using a leaky bucket concept. The error count returned when this register is read is not an absolute count over time, but the sum of errors during a current specified time period plus half of the accumulated errors from past time periods. When a time period expires (determined by the Spare Control register – D0, F0, 90 - 93h), the sum of the current time period accumulated errors and a value equal to half of the past accumulated errors is retained. Half of this registered error value will be added to the errors accumulated during the next time period. This method is employed, because it is not the absolute number of errors that is most interesting, but the rate that errors occur. When this register is written, the counter holding the number of errors with the current time period is updated. Because of this described structure, reading back the register will only return the same value written if no time period has expired between the write and the read. A write to this register does clears out any error residue that may exist from past time periods. The settings in the DRM register will map the logical DIMMs to the physical DIMMs. For DDR2 DIMMs, these counters are on a per rank basis. For single rank DDR2 DIMMs, this rank basis still corresponds to a DIMM basis, since it is a single rank within the DIMM. It is only for dual rank DDR2 DIMMs that these counters cannot be treated as DIMM counters, and only as rank counters

The bits in this register are sticky through reset.

Table 257. Offset B0 - B1h: DRAM_SEC_D0A – DRAM SEC Logical DIMM 0 Counter Register

Device: 0	**Function:** 1
Offset: B0 - B1h	**Size:** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
15:00	SEC_D0A	**SEC error counter for DIMM0:** DIMM0 SEC Count.	0000h	RW

13.2.1.49 Offset B2 - B3h: DRAM_DED_D0A – DRAM DED Logical DIMM 0 Counter Register

Counter for DED errors occurring for logical DIMM0. The functionality of this counter is described in Section 13.2.1.48. The bits in this register are sticky through reset.

Table 258. Offset B2 - B3h: DRAM_DED_D0A – DRAM DED Logical DIMM 0 Counter Register

Device: 0 ***Function:*** 1
Offset: *B2 - B3h* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:00	DED_D0A	**DED error counter for DIMM0:**DIMM0DED Count.	0000h	RW

13.2.1.50 Offset B4 - B5h: DRAM_SEC_D1A – DRAM SEC Logical DIMM 1 Counter Register

Counter for SEC errors occurring for logical DIMM1 of channel A. The functionality of this counter is described in Section 13.2.1.48. The bits in this register are sticky through reset.

Table 259. Offset B4 - B5h: DRAM_SEC_D1A – DRAM SEC Logical DIMM 1 Counter Register

Device: 0 ***Function:*** 1
Offset: *B4 - B5h* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:00	SEC_D1A	**SEC error counter for DIMM1:**DIMM1SEC Count.	0000h	RW

13.2.1.51 Offset B6 - B7h: DRAM_DED_D1A – DRAM DED Logical DIMM 1 Counter Register

Counter for DED errors occurring for logical DIMM1. The functionality of this counter is described in Section 13.2.1.48. The bits in this register are sticky through reset.

Table 260. Offset B6 - B7h: DRAM_DED_D1A – DRAM DED Logical DIMM 1 Counter Register

Device: 0 ***Function:*** 1
Offset: *B6 - B7h* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:00	DED_D1A	**DED error counter for DIMM1:** DIMM1DED Count.	0000h	RW

13.2.1.52 Offset B8 - B9h: DRAM_SEC_D2A – DRAM SEC Logical DIMM 2 Counter Register

Counter for SEC errors occurring for logical DIMM2. The functionality of this counter is described in Section 13.2.1.48. The bits in this register are sticky through reset.

Table 261. Offset B8 - B9h: DRAM_SEC_D2A – DRAM SEC Logical DIMM 2 Counter Register

Device: 0 ***Function:*** 1
Offset: *B8 - B9h* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:00	SEC_D2A	**SEC error counter for DIMM2:** DIMM2SEC Count.	0000h	RW

13.2.1.53 Offset BA - BBh: DRAM_DED_D2A – DRAM DED Logical DIMM 2 Counter Register

Counter for DED errors occurring for logical DIMM2. The functionality of this counter is described in Section 13.2.1.48. The bits in this register are sticky through reset.

Table 262. Offset BA - BBh: DRAM_DED_D2A – DRAM DED Logical DIMM 2 Counter Register

Device: 0 ***Function:*** 1
Offset: *BA - BBh* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:00	DED_D2A	**DED error counter for DIMM2:**DIMM2DED Count.	0000h	RW

13.2.1.54 Offset BC - BDh: DRAM_SEC_D3A – DRAM SEC Logical DIMM 3 Counter Register

Counter for SEC errors occurring for logical DIMM3. The functionality of this counter is described in Section 13.2.1.48. The bits in this register are sticky through reset.

Table 263. Offset BC - BDh: DRAM_SEC_D3A – DRAM SEC Logical DIMM 3 Counter Register

Device: 0 ***Function:*** 1
Offset: *BC - BDh* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:00	SEC_D3A	**SEC error counter for DIMM3:** DIMM3SEC Count.	0000h	RW

13.2.1.55 Offset BE - BFh: DRAM_DED_D3A – DRAM DED Logical DIMM 3 Counter Register

Counter for DED errors occurring for logical DIMM3. The functionality of this counter is described in Section 13.2.1.48. The bits in this register are sticky through reset.

Table 264. Offset BE - BFh: DRAM_DED_D3A – DRAM DED Logical DIMM 3 Counter Register

Device: 0	***Function:*** 1
Offset: *BE - BFh*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:00	DED_D3A	**DED error counter for DIMM3:**DIMM3DED Count.	0000h	RW

13.2.1.56 Offset C2 - C3h: THRESH_DED – Threshold for DEDs Register

This register is the threshold compare value for DED errors. An Error Threshold Detect is signaled if any of the DED counters exceeds the value programmed into this register. The bits in this register are sticky through reset.

Table 265. Offset C2 - C3h: THRESH_DED – Threshold for DEDs Register

Device: 0	***Function:*** 1
Offset: *C2 - C3h*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:00	THRESH_DED	**DED Error Threshold:** Threshold compare value for DED errors.	0000h	RW

13.2.1.57 Offset C4 - C5h: DRAM_SECF_SYNDROME – DRAM First Single Error Correct Syndrome Register

Syndrome for correctable errors occurring in the memory system. The contents of this register are set when correctable error bit is being set in the DRAM_FERR register (bit 0). Syndrome is always logged for QW0/2 or QW1/3 pairs (block), if transferring the lower half of the cache line, and logged for QW4/6 or QW5/7 if transferring the upper half of the cache line. ECC is checked ½ cacheline at a time. The syndrome logged in this register is for the lowest ordered QW pair. For example: If both QW0/2 and QW1/3 have correctable errors, the syndrome stored is for QW0/2. The syndrome is stored in bits 15:08. A syndrome indicates error when it is a non-zero value. Refer to the Intel® 3100 Chipset *BIOS Specification* for more complete details. The bits in this register are sticky through reset.

Table 266. Offset C4 - C5h: DRAM_SECF_SYNDROME – DRAM First Single Error Correct Syndrome Register

Device: 0	**Function:** 1
Offset: C4 - C5h	**Size:** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
15:00	SECF_SYND	**ECC Syndrome for first correctable error**. Because only hardware writes to this register, it is read-only. **SEC mode** Bits 15:08 Mem Channel Bits 07:00 Reserved	0000h	RO

13.2.1.58 Offset C6 - C7h: DRAM_SECN_SYNDROME – DRAM Next Single Error Correct Syndrome Register

Syndrome for next correctable error occurring in the memory system. The contents of this register correspond to the correctable error bit being set in the DRAM_NERR register. Refer to the IMCH *BIOS Specification* for more complete details. The bits in this register are sticky through reset.

Table 267. Offset C6 - C7h: DRAM_SECN_SYNDROME – DRAM Next Single Error Correct Syndrome Register

Device: 0	**Function:** 1
Offset: C6 - C7h	**Size:** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
15:00	SECN_SYND	**ECC Syndrome for DRAM_NERR correctable error indicated by DRAM NERR register:** Because only hardware writes to this register, it is Read-Only. Details are in Section 13.2.1.57.	0000h	RO

13.2.1.59 Offset C8 - CBh: DRAM_SECN_ADD – DRAM Next Single Bit Error Correct Address Register

Captures the address of the next SEC error (either normal or scrub read) occurring in the memory system. The value in this register is only valid if the Correctable Read Memory Error bit in the DRAM_NERR register has been set. The bits in this register are sticky through reset.

Table 268. Offset C8 - CBh: DRAM_SECN_ADD – DRAM Next Single Bit Error Correct Address Register

Device: 0	***Function:*** 1
Offset: *C8 - CBh*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
31	Reserved	Reserved	0b	
30:02	SECNADD	**Next Correctable Error Address:** This field contains address bits 35:12 for the next correctable error. This field is set by hardware when the Correctable Read Memory Error bit in the DRAM_SERR register is set. This value represents a physical address. This field can only be reset by a PWRGD reset.	000_0000h	RO
01:00	Reserved	Reserved	0b	

13.2.1.60 Offset DC - DDh: DIMMTHREX – DIMM Threshold Exceeded Register

Preserves knowledge of DIMM error thresholds exceeded. The bits in this register are sticky through reset.

Table 269. Offset DC - DDh: DIMMTHREX – DIMM Threshold Exceeded Register (Sheet 1 of 2)

Device: 0	***Function:*** 1
Offset: *DC - DDh*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:08	Reserved	**Reserved**	0h	
07	ADIMM3_DED	**Logical DIMM 3 DED threshold Status:** Logical DIMM 3 DED Threshold Status. This bit is sticky through reset. Software can clear this bit by writing a 1 to the bit location. 0 = Threshold not exceeded 1 = Threshold exceeded	0b	RWC
06	ADIMM2_DED	**Logical DIMM 2 DED threshold Status:** Logical DIMM 2 DED Threshold Status. This bit is sticky through reset. Software can clear this bit by writing a 1 to the bit location. 0 = Threshold not exceeded 1 = Threshold exceeded	0b	RWC
05	ADIMM1_DED	**Logical DIMM 1 DED threshold Status:** Logical DIMM 1 DED Threshold Status. This bit is sticky through reset. Software can clear this bit by writing a 1 to the bit location. 0 = Threshold not exceeded 1 = Threshold exceeded	0b	RWC

Table 269. Offset DC - DDh: DIMMTHREX – DIMM Threshold Exceeded Register (Sheet 2 of 2)

Device: 0	**Function:** 1
Offset: DC - DDh	**Size:** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
04	ADIMM0_DED	**Logical DIMM 0 DED threshold Status:** Logical DIMM 0 DED Threshold Status. This bit is sticky through reset. Software can clear this bit by writing a 1 to the bit location. 0 = Threshold not exceeded 1 = Threshold exceeded	0b	RWC
03	ADIMM3_SEC	**Logical DIMM 3 SEC threshold Status:** Logical DIMM 3 SEC Threshold Status. This bit is sticky through reset. Software can clear this bit by writing a 1 to the bit location. 0 = Threshold not exceeded 1 = Threshold exceeded	0b	RWC
02	ADIMM2_SEC	**Logical DIMM 2 SEC threshold Status:** Logical DIMM 2 SEC Threshold Status. This bit is sticky through reset. Software can clear this bit by writing a 1 to the bit location. 0 = Threshold not exceeded 1 = Threshold exceeded	0b	RWC
01	ADIMM1_SEC	**Logical DIMM 1 SEC threshold Status:** Logical DIMM 1 SEC Threshold Status. This bit is sticky through reset. Software can clear this bit by writing a 1 to the bit location. 0 = Threshold not exceeded 1 = Threshold exceeded	0b	RWC
00	ADIMM0_SEC	**Logical DIMM 0 SEC threshold Status:** Logical DIMM 0 SEC Threshold Status. This bit is sticky through reset. Software can clear this bit by writing a 1 to the bit location. 0 = Threshold not exceeded 1 = Threshold exceeded	0b	RWC

13.2.1.61 Offset E0h - E3h: HERRCTL – Host Error Control Register

This register controls the way in which the IMCH handles parity errors detected on incoming data streams into the IMCH core from the FSB.

Table 270. Offset E0h - E3h: HERRCTL – Host Error Control Register (Sheet 1 of 2)

Device: 0	**Function:** 1
Offset: E0h - E3h	**Size:** 32 bit
Default Value: 0020_0000h	

Bits	Name	Description	Reset Value	Access
31:22	Reserved	Reserved	00h	
21	Reserved	Reserved.	1b	

Table 270. Offset E0h - E3h: HERRCTL – Host Error Control Register (Sheet 2 of 2)

Device: 0	**Function:** 1
Offset: E0h - E3h	**Size:** 32 bit
Default Value: 0020_0000h	

Bits	Name	Description	Reset Value	Access
20:19	Reserved	Reserved	00b	
18	EnDP	**Enable data poisoning:** This bit controls whether or not the IMCH marks data as "poisoned" when a parity error is detected on incoming data from the FSB. 0 = Error checking disabled. 1 = Error checking enabled. Incoming data with parity errors are marked as "poisoned" before being sent on towards its destination.	0b	RW
17:00	Reserved	Reserved.	0000h	

13.2.1.62 Offset E8h - EBh: BERRCTL – Buffer Error Control Register

This register enables the injection of errors on data read out of the posted write buffer. The lower 16 bits are the corresponding flip parity bits for the cacheline of data. The upper bits in the register are for the use and control of the associated flip parity bits.

Table 271. Offset E8h - EBh: BERRCTL – Buffer Error Control Register

Bits	Name	Description	Reset Value	Access
31:19	Reserved	Reserved	00h	
18	EnDP	**Enable/Disable data poisoning:** Error Injection is possible regardless of this bit setting. 0 = Data Poisoning disabled - Errors won't be propagated, only good parity is generated. 1 = Data Poisoning enabled. Errors will be propagated.	0b	RW
17:00	Reserved	Reserved	0000h	

13.2.1.63 Offset ECh - EFh: DERRCTL – DRAM Error Control Register

This register controls the IMCH handling of errors on incoming data streams into the IMCH core from the DRAM interface.

Table 272. Offset ECh - EFh: DERRCTL – DRAM Error Control Register

Device: 0	***Function:*** 1
Offset: *ECh - EFh*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
31:19	Reserved	Reserved	00h	
18	EnDP	**Enable / Disable data poisoning for incoming data:** This bit controls whether or not the IMCH marks data as "poisoned" when a parity error is detected on incoming data from the DRAM I/F. 0 = Errors are not propagated, only good internal parity generated. 1 = Error Poisoning Enabled. Incoming data with parity errors are marked as "poisoned" before being sent on towards its destination when in either 72-bit ECC mode via the DRC register.	0b	RW
17:00	Reserved	Reserved	0000h	

13.3 Device 1, Function 0: EDMA Registers

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 273. EDMA Configuration Register Map

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	01h	VID	Vendor Identification Register	8086h	RO
02h	03h	DID	Device Identification Register	35B5h	RO
04h	05h	PCICMD	PCI Command Register	0000h	RO, RW
06h	07h	PCISTS	PCI Status Register	0010h	RO, RWC
08h	08h	RID	Revision Identification Register	00h	RO
0Ah	0Ah	SUBC	Sub-Class Code Register	80h	RO
0Bh	0Bh	BCC	Base Class Code Register	08h	RO
0Eh	0Eh	HDR	Header Type Register	00h	RO
10h	13h	EDMALBAR	EDMA Low Base Address Register	0000_0000h	RO, RW
2Ch	2Dh	SVID	Subsystem Vendor Identification Register	0000h	RWO
2Eh	2Fh	SID	Subsystem Identification Register	0000h	RWO
34h	34h	CAPPTR	Capabilities Pointer Register	B0h	RO
3Ch	3Ch	INTRLINE	Interrupt Line Register	00h	RW
3Dh	3Dh	INTRPIN	Interrupt Pin Register	01h	RO
40h	40h	EDMACTL	EDMA Control Register	08h	RO, RW
80h	83h	EDMA_FERR	EDMA First Error Register	0000_0000h	RO, RWC
84h	87h	EDMA_NERR	EDMA Next Error Register	0000_0000h	RO, RWC
88h	88h	EDMA_EMASK	EDMA Error Mask Register	00h	RW
A0h	A0h	EDMA_SCICMD	EDMA SCI Command Register	00h	RW
A4h	A4h	EDMA_SMICMD	EDMA SMI Command Register	00h	RW
A8h	A8h	EDMA_SERRCMD	EDMA SERR Command Register	00h	RW
ACh	ACh	EDMA_MCERRCMD	EDMA MCERR Command Register	00h	RW
B0h	B3h	MSICR	MSI Control Register	0002_0005h	RO, RW
B4h	B7h	MSIAR	MSI Address Register	FEE0_0000h	RW
B8h	B9h	MSIDR	MSI Data Register	0000h	RO, RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

13.3.1 Register Details

13.3.1.1 Offset 00h - 01h: VID – Vendor Identification Register

The VID Register contains the vendor identification number. This 16-bit register combined with the Device Identification Register uniquely identifies a PCI device.

Table 274. Offset 00h - 01h: VID – Vendor Identification Register

Device: 1	***Function:*** 0
Offset: *00 - 01h*	***Size:*** *16 bit*
Default Value: *8086h*	

Bits	Name	Description	Reset Value	Access
15:00	VID	**Vendor Identification:** This register field contains the PCI standard identification for Intel 8086h.	8086h	RO

13.3.1.2 Offset 02h - 03h: DID – Device Identification Register

This 16-bit register combined with the Vendor Identification register uniquely identifies any PCI device.

Table 275. Offset 02h - 03h: DID – Device Identification Register

Device: 1	***Function:*** 0
Offset: *02 - 03h*	***Size:*** *16 bit*
Default Value: *35B5h*	

Bits	Name	Description	Reset Value	Access
15:00	DID	**Device Identification Number:** This is a 16-bit value assigned to the IMCH EDMA controller Function 0.	35B5h	RO

13.3.1.3 Offset 04h - 05h: PCICMD – PCI Command Register

Table 276. Offset 04h - 05h: PCICMD – PCI Command Register (Sheet 1 of 2)

Device: 1	***Function:*** 0
Offset: *04 - 05h*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:11	Reserved	Reserved	00h	
10	INTxAD	**INTx Assertion DIsable:** Controls the ability of a device to generate INTx interrupt messages. This bit only applies to legacy interrupts and not MSIs. 0 = Devices are permitted to generate INTx interrupt messages. 1 = Devices are prevented from generating INTx interrupt messages.	0b	RW
09	FB2B	**Fast Back-to-Back Enable:** Not Applicable-hardwired to 0.	0b	RO

Table 276. Offset 04h - 05h: PCICMD – PCI Command Register (Sheet 2 of 2)

Device: 1	***Function:*** 0
Offset: *04 - 05h*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
08	SERRE	**SERR Enable:** This bit is a global enable bit for Device 1 SERR messaging. The EDMA does not have an SERR# signal. The EDMA communicates the SERR condition by sending an SERR message. 0 = Disable. SERR message is not generated for Device 1. 1 = Enable. Generate SERR messages for specific Device 1 error conditions that are individually enabled in the EDMA_SERRCMD register. The error status is reported in the EDMA_FERR, EDMA_NERR and PCISTS registers. ***Note:*** This bit only controls SERR messaging for Device 1. Devices 0 and 2–7 have their own SERR bits to control error reporting for error conditions occurring on their respective devices. The control bits are used in a logical OR manner to enable the SERR HI message mechanism	0b	RW
07:02	Reserved	Reserved	00h	
01	MAE	**Memory Access Enable:** 0 = Device 1 memory space is disabled 1 = Enable access to the EDMA Controller Low Base Address Register	0b	RW
00	Reserved	Reserved	0b	

13.3.1.4 Offset 06h - 07h: PCISTS – PCI Status Register

Table 277. Offset 06h - 07h: PCISTS – PCI Status Register (Sheet 1 of 2)

Device: 1	***Function:*** 0
Offset: *06 - 07h*	***Size:*** *16 bit*
Default Value: *0010h*	

Bits	Name	Description	Reset Value	Access
15	Reserved	Reserved	0b	
14	SSE	**Signaled System Error:** 0 = Device 1 has not generated a SERR. Software clears this bit by writing a 1 to the bit location. 1 = Indicates Device 1 generated an SERR message for any enabled Device 2 error condition. Device 1 error conditions are enabled in the PCICMD and EDMA_SERRCMD registers. Device 1 error flags are read/reset from the PCISTS, EDMA_FERR, or EDMA_NERR registers.	0b	RWC
13:05	Reserved	Reserved	000h	

Table 277. Offset 06h - 07h: PCISTS – PCI Status Register (Sheet 2 of 2)

Device: 1	**Function:** 0
Offset: 06 - 07h	**Size:** 16 bit
Default Value: 0010h	

Bits	Name	Description	Reset Value	Access
04	CLIST	**Capability List:** This bit is set to 1 to indicate to the configuration software that this device/function implements a list of new capabilities. A list of new capabilities is accessed via register CAPPTR at configuration address offset 34h.	1b	RO
03	INTx	**INTx Status:** Indicates that an INTx interrupt is pending internal to the device. The interrupt assertion disable bit has no affect on the setting of this bit. This bit is not set for an MSI. 0 = An INTx interrupt is NOT pending 1 = An INTx interrupt is pending internal to the device	0b	RO
02:00	Reserved	Reserved	0h	

13.3.1.5 Offset 08h: RID – Revision Identification Register

This register contains the revision number of Device 1.

Table 278. Offset 08h: RID – Revision Identification Register

Device: 1	**Function:** 0
Offset: 08h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07:00	RID	**Revision Identification Number:** This is an 8-bit value that indicates the revision identification number for Device 1. This number must always be the same as the RID for Device 0, Function 0.	00h	RO

13.3.1.6 Offset 0Ah: SUBC – Sub-Class Code Register

Table 279. Offset 0Ah: SUBC – Sub-Class Code Register

Device: 1	**Function:** 0
Offset: 0Ah	**Size:** 8 bit
Default Value: 80h	

Bits	Name	Description	Reset Value	Access
07:00	SUBC	**Sub-Class Code:** This is an 8-bit value that indicates the category of other (non-specific) system peripheral.	80h	RO

13.3.1.7 Offset 0Bh: BCC – Base Class Code Register

Table 280. Offset 0Bh: BCC – Base Class Code Register

Device: 1	***Function:*** 0
Offset: *0Bh*	***Size:*** *8 bit*
Default Value: *08h*	

Bits	Name	Description	Reset Value	Access
07:00	BASEC	**Base Class Code:** This is an 8-bit value that indicates the Base Class Code for a system peripheral.	08h	RO

13.3.1.8 Offset 0Eh: HDR – Header Type Register

Table 281. Offset 0Eh: HDR – Header Type Register

Device: 1	***Function:*** 0
Offset: *0Eh*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:00	HDR	**PCI Header:** This value indicates the header type of the IMCH Device 1. 00h = single-function device.	00h	RO

13.3.1.9 Offset 10h - 13h: EDMALBAR – EDMA Low Base Address Register

Table 282. Offset 10h - 13h: EDMALBAR – EDMA Low Base Address Register

Device: 1	***Function:*** 0
Offset: *10 - 13h*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
31:12	UPBITS	**Upper Programmable Base Address:** These bits are part of the System Memory MMR region, normally set by configuration software to locate the base address of the region.	00000h	RW
11:04	LOWBITS	**Lower Bits:** These bits are hardwired to 0. This forces the size of the memory region to be 4 Kbyte.	000h	RO
03	PF	**Prefetchable:** This bit is hardwired to 0 to indicate that the System Memory MMR region is NON-Prefetchable.	0b	RO
02:01	TYPE	**Addressing Type:** These bits determine the addressing type and they are hardwired to 00 to indicate that the address range defined by the upper bits of this register can be located anywhere in the 32-bit address space as per the PCI specification for base address registers.	00b	RO
00	MSPACE	**Memory Space Indicator:** This bit is hardwired to 0 to identify the MMR range as a memory range as per the specification for PCI base address registers.	0b	RO

13.3.1.10 Offset 2Ch - 2Dh: SVID – Subsystem Vendor Identification Register

This value is used to identify the vendor of the subsystem.

Table 283. Offset 2Ch - 2Dh: SVID – Subsystem Vendor Identification Register

Device: 1	***Function:*** 0
Offset: *2C - 2Dh*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:00	SUBVID	**Subsystem Vendor ID:** This field must be programmed during boot-up to indicate the vendor of the system board. When any byte or combination of bytes of this register is written, the register value locks and cannot be further updated.	0000h	RWO

13.3.1.11 Offset 2Eh - 2Fh: SID – Subsystem Identification Register

This value is used to identify a particular subsystem.

Table 284. Offset 2Eh - 2Fh: SID – Subsystem Identification Register

Device: 1	***Function:*** 0
Offset: *2E - 2Fh*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:00	SUBID	**Subsystem ID:** This field must be programmed during BIOS initialization. When any byte or combination of bytes of this register is written, the register value locks and cannot be further updated.	0000h	RWO

13.3.1.12 Offset 34h: CAPPTR – Capabilities Pointer Register

The CAPPTR provides the offset that is the pointer to the location where the first set of capabilities registers is located.

Table 285. Offset 34h: CAPPTR – Capabilities Pointer Register

Device: 1	***Function:*** 0
Offset: *34h*	***Size:*** *8 bit*
Default Value: *B0h*	

Bits	Name	Description	Reset Value	Access
07:00	CAP_PTR	**Capabilities Pointer:** Pointer to first capabilities structure.	B0h	RO

13.3.1.13 Offset 3Ch: INTRLINE – Interrupt Line Register

Table 286. Offset 3Ch: INTRLINE – Interrupt Line Register

Device: 1	**Function:** 0
Offset: 3Ch	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07:00	INTRLINE	**Interrupt Connection:** BIOS writes the interrupt routing information to this register to indicate which input of the interrupt controller is connected with this device.	00h	RW

13.3.1.14 Offset 3Dh: INTRPIN – Interrupt Pin Register

Table 287. Offset 3Dh: INTRPIN – Interrupt Pin Register

Device: 1	**Function:** 0
Offset: 3Dh	**Size:** 8 bit
Default Value: 01h	

Bits	Name	Description	Reset Value	Access
07:00	INTRPIN	**Interrupt Pin:** Set to 01h to specify that EDMA always uses INTA# as its interrupt pin.	01h	RO

13.3.1.15 Offset 40h: EDMACTL – EDMA Control Register

This register defines global operation of the EDMA channels.

Table 288. Offset 40h: EDMACTL – EDMA Control Register

Device: 1	**Function:** 0
Offset: 40h	**Size:** 8 bit
Default Value: 08h	

Bits	Name	Description	Reset Value	Access
07	EDMAEN	**EDMA Engine Enable:** 0 = EDMA Engine is disabled. 1 = EDMA Engine is enabled to do transfers. Whether the bit is set or clear, the registers can be programmed.	0b	RW
06:04	Reserved	Reserved	000b	
03	Reserved	This bit MUST be set to 1 by BIOS for proper operation.	1b	
02:01	Reserved	Reserved	00b	
00	PARITYEN	**Disable abort on data parity error:** 0 = Controller aborts on data parity error 1 = Controller does not abort on data parity errors, but logs the event in the appropriate EDMA FERR/NERR bit	0b	RW

13.3.1.16 Offset 80h - 83h: EDMA_FERR – EDMA First Error Register

This register captures the first occurrence of errors on a EDMA channel basis. This register only designates which channel had an error. Once an error has been captured for a given channel, this register is locked and needs to be written with ones to clear it. All EDMA errors are considered non-fatal because they cause the EDMA engine to stop further processing, thus avoiding any data corruption. The errors are fatal to the process, but not to the system. This register is sticky through reset.

Table 289. Offset 80h - 83h: EDMA_FERR – EDMA First Error Register (Sheet 1 of 2)

Device: 1 ***Function:*** 0
Offset: *80h - 83h* ***Size:*** *32-bit*
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31	Channel 3 NDAR Addressing Error	The descriptor pointer in the next descriptor address register is of incorrect type or range for EDMA channel 3. This includes above TOM, not in a memory range, and above available address space. (NON-FATAL)	0b	RWC
30	Channel 3 NDAR Alignment Error	The descriptor pointer in the next descriptor address register is not aligned to an eight double-word boundary for EDMA channel 3. (NON-FATAL)	0b	RWC
29	Channel 3 Source Address Error	The source address does not comply with the source type or range for EDMA channel 3. (NON-FATAL)	0b	RWC
28	Reserved	Reserved	0b	
27	Channel 3 Destination Address Error	The destination address does not comply with the destination type or range for EDMA channel 3. (NON-FATAL)	0b	RWC
26	Reserved	Reserved	0b	
25	Channel 3 Parity Error	Data parity Error in reading source data from system memory for EDMA channel 3. (NON-FATAL)	0b	RWC
24	Channel 3 Write Error	Received write to RO descriptor registers for EDMA channel 3. (NON-FATAL)	0b	RWC
23	Channel 2 NDAR Addressing Error	The descriptor pointer in the next descriptor address register is of incorrect type or range for EDMA channel 2. This includes above TOM, not in a memory range, and above available address space. (NON-FATAL)	0b	RWC
22	Channel 2 NDAR Alignment Error	The descriptor pointer in the next descriptor address register is not aligned to an eight double-word boundary for EDMA channel 2. (NON-FATAL)	0b	RWC
21	Channel 2 Source Address Error	The source address does not comply with the source type or range for EDMA channel 2. (NON-FATAL)	0b	RWC
20	Reserved	Reserved	0b	
19	Channel 2 Destination Address Error	The destination address does not comply with the destination type or range for EDMA channel 2. (NON-FATAL)	0b	RWC
18	Reserved	Reserved	0b	
17	Channel 2 Parity Error	Data parity Error in reading source data from system memory for EDMA channel 2. (NON-FATAL)	0b	RWC
16	Channel 2 Write Error	Received write to RO descriptor registers for EDMA channel 2. (NON-FATAL)	0b	RWC
15	Channel 1 NDAR Addressing Error	The descriptor pointer in the next descriptor address register is of incorrect type or range for EDMA channel 1. This includes above TOM, not in a memory range, and above available address space. (NON-FATAL)	0b	RWC
14	Channel 1 NDAR Alignment Error	The descriptor pointer in the next descriptor address register is not aligned to an eight double-word boundary for EDMA channel 1. (NON-FATAL)	0b	RWC
13	Channel 1 Source Address Error	The source address does not comply with the source type or range for EDMA channel 1. (NON-FATAL)	0b	RWC

Table 289. Offset 80h - 83h: EDMA_FERR – EDMA First Error Register (Sheet 2 of 2)

Device: 1 **Function:** 0
Offset: 80h - 83h **Size:** 32-bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
12	Reserved	Reserved	0b	
11	Channel 1 Destination Address Error	The destination address does not comply with the destination type or range for EDMA channel 1. (NON-FATAL)	0b	RWC
10	Reserved	Reserved	0b	
9	Channel 1 Parity Error	Data parity Error in reading source data from system memory for EDMA channel 1. (NON-FATAL)	0b	RWC
8	Channel 1 Write Error	Received write to RO descriptor registers for EDMA channel 1. (NON-FATAL)	0b	RWC
7	Channel 0 NDAR Addressing Error	The descriptor pointer in the next descriptor address register is of incorrect type or range for EDMA channel 0. This includes above TOM, not in a memory range, and above available address space. (NON-FATAL)	0b	RWC
6	Channel 0 NDAR Alignment Error	The descriptor pointer in the next descriptor address register is not aligned to an eight double-word boundary for EDMA channel 0. (NON-FATAL)	0b	RWC
5	Channel 0 Source Address Error	The source address does not comply with the source type or range for EDMA channel 0. (NON-FATAL)	0b	RWC
4	Reserved	Reserved	0b	
3	Channel 0 Destination Address Error	The destination address does not comply with the destination type or range for EDMA channel 0. (NON-FATAL)	0b	RWC
2	Reserved	Reserved	0b	
1	Channel 0 Parity Error	Data parity Error in reading source data from system memory for EDMA channel 0. (NON-FATAL)	0b	RWC
0	Channel 0 Write Error	Received write to RO descriptor registers for EDMA channel 0. (NON-FATAL)	0b	RWC

13.3.1.17 Offset 84h - 87h: EDMA_NERR – EDMA Next Error Register

This register captures EDMA channel errors after the FERR register is locked. This register accumulates all subsequent errors for the EDMA channels. See Table 289 for bit definitions. This register is sticky through reset.

Table 290. Offset 84h - 87h: EDMA_NERR – EDMA Next Error Register (Sheet 1 of 3)

Device: 1 **Function:** 0
Offset: 84h -87h **Size:** 32-bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31	Channel 3 NDAR Addressing Error	The descriptor pointer in the next descriptor address register is of incorrect type or range for EDMA channel 3. This includes above TOM, not in a memory range, and above available address space. (NON-FATAL)	0b	RWC
30	Channel 3 NDAR Alignment Error	The descriptor pointer in the next descriptor address register is not aligned to an eight double-word boundary for EDMA channel 3. (NON-FATAL)	0b	RWC
29	Channel 3 Source Address Error	The source address does not comply with the source type or range for EDMA channel 3. (NON-FATAL)	0b	RWC

Table 290. Offset 84h - 87h: EDMA_NERR – EDMA Next Error Register (Sheet 2 of 3)

Device: 1 **Function:** 0
Offset: 84h -87h **Size:** 32-bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
28	Reserved	Reserved	0b	
27	Channel 3 Destination Address Error	The destination address does not comply with the destination type or range for EDMA channel 3. (NON-FATAL)	0b	RWC
26	Reserved	Reserved	0b	
25	Channel 3 Parity Error	Data parity Error in reading source data from system memory for EDMA channel 3. (NON-FATAL)	0b	RWC
24	Channel 3 Write Error	Received write to RO descriptor registers for EDMA channel 3. (NON-FATAL)	0b	RWC
23	Channel 2 NDAR Addressing Error	The descriptor pointer in the next descriptor address register is of incorrect type or range for EDMA channel 2. This includes above TOM, not in a memory range, and above available address space. (NON-FATAL)	0b	RWC
22	Channel 2 NDAR Alignment Error	The descriptor pointer in the next descriptor address register is not aligned to an eight double-word boundary for EDMA channel 2. (NON-FATAL)	0b	RWC
21	Channel 2 Source Address Error	The source address does not comply with the source type or range for EDMA channel 2. (NON-FATAL)	0b	RWC
20	Reserved	Reserved	0b	
19	Channel 2 Destination Address Error	The destination address does not comply with the destination type or range for EDMA channel 2. (NON-FATAL)	0b	RWC
18	Reserved	Reserved	0b	
17	Channel 2 Parity Error	Data parity Error in reading source data from system memory for EDMA channel 2. (NON-FATAL)	0b	RWC
16	Channel 2 Write Error	Received write to RO descriptor registers for EDMA channel 2. (NON-FATAL)	0b	RWC
15	Channel 1 NDAR Addressing Error	The descriptor pointer in the next descriptor address register is of incorrect type or range for EDMA channel 1. This includes above TOM, not in a memory range, and above available address space. (NON-FATAL)	0b	RWC
14	Channel 1 NDAR Alignment Error	The descriptor pointer in the next descriptor address register is not aligned to an eight double-word boundary for EDMA channel 1. (NON-FATAL)	0b	RWC
13	Channel 1 Source Address Error	The source address does not comply with the source type or range for EDMA channel 1. (NON-FATAL)	0b	RWC
12	Reserved	Reserved	0b	
11	Channel 1 Destination Address Error	The destination address does not comply with the destination type or range for EDMA channel 1. (NON-FATAL)	0b	RWC
10	Reserved	Reserved	0b	
9	Channel 1 Parity Error	Data parity Error in reading source data from system memory for EDMA channel 1. (NON-FATAL)	0b	RWC
8	Channel 1 Write Error	Received write to RO descriptor registers for EDMA channel 1. (NON-FATAL)	0b	RWC
7	Channel 0 NDAR Addressing Error	The descriptor pointer in the next descriptor address register is of incorrect type or range for EDMA channel 0. This includes above TOM, not in a memory range, and above available address space. (NON-FATAL)	0b	RWC
6	Channel 0 NDAR Alignment Error	The descriptor pointer in the next descriptor address register is not aligned to an eight double-word boundary for EDMA channel 0. (NON-FATAL)	0b	RWC
5	Channel 0 Source Address Error	The source address does not comply with the source type or range for EDMA channel 0. (NON-FATAL)	0b	RWC

Table 290. Offset 84h - 87h: EDMA_NERR – EDMA Next Error Register (Sheet 3 of 3)

Device: 1	***Function:*** 0
Offset: *84h -87h*	***Size:*** *32-bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
4	Reserved	Reserved	0b	
3	Channel 0 Destination Address Error	The destination address does not comply with the destination type or range for EDMA channel 0. (NON-FATAL)	0b	RWC
2	Reserved	Reserved	0b	
1	Channel 0 Parity Error	Data parity Error in reading source data from system memory for EDMA channel 0. (NON-FATAL)	0b	RWC
0	Channel 0 Write Error	Received write to RO descriptor registers for EDMA channel 0. (NON-FATAL)	0b	RWC

13.3.1.18 Offset 88h: EDMA_EMASK – EDMA Error Mask Register

This register masks the unit errors from being recognized and therefore not logged at the unit or global level and no interrupt/messages are generated. All channels are expected to use the same reporting structure, so only one 8-bit register is implemented. These bits are sticky through reset.

Table 291. Offset 88h: EDMA_EMASK – EDMA Error Mask Register (Sheet 1 of 2)

Device: 1	***Function:*** 0
Offset: *88h*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07	DSCPEM	**Descriptor Address Type/Range Error Mask:** Mask bit for error bit 7, 15, 23, and 31 of EDMA_FERR and EDMA_NERR. This bit is sticky through reset. 0 = Allow descriptor address type/range error logging and signaling. 1 = Mask descriptor address type/range error logging and signaling.	0b	RW
06	DSCPAE	**Descriptor Address Alignment Error:** Mask bit for error bit 6, 14, 22, and 30 of EDMA_FERR and EDMA_NERR. This bit is sticky through reset. 0 = Allow descriptor address alignment error logging and signaling. 1 = Mask descriptor address alignment error logging and signaling.	0b	RW
05	SRCEM	**Source Address Type/Range Error:** Mask bit for error bit 5, 13, 21, and 29 of EDMA_FERR and EDMA_NERR. This bit is sticky through reset. 0 = Allow source address type/range error logging and signaling. 1 = Mask source address type/range error logging and signaling.	0b	RW
04	Reserved	**Reserved**	0b	

Table 291. Offset 88h: EDMA_EMASK – EDMA Error Mask Register (Sheet 2 of 2)

Device: 1	**Function:** 0
Offset: 88h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
03	DESTEM	**Destination Address Type/Range Error:** Mask bit for error bit 3, 11, 19, and 27 of EDMA_FERR and EDMA_NERR. This bit is sticky through reset. 0 = Allow destination address type/range error logging and signaling. 1 = Mask destination address type/range error logging and signaling.	0b	RW
02	Reserved	**Reserved**	0b	
01	MDPARERR	**Memory Data Parity Error:** Mask bit for error bit 1, 9, 17, and 25 of EDMA_FERR and EDMA_NERR. This bit is sticky through reset. 0 = Allow memory data parity error logging and signaling. 1 = Mask memory data parity error logging and signaling.	0b	RW
00	IWERR	**Illegal Write Error:** Mask bit for error bit 0, 8, 16, and 24 of EDMA_FERR and EDMA_NERR. This bit is sticky through reset. 0 = Allow illegal write error logging and signaling. 1 = Mask illegal write error logging and signaling.	0b	RW

13.3.1.19 Offset A0h: EDMA_SCICMD – EDMA SCI Command Register

This register enables various errors to generate an SCI special cycle to the IICH. When an error flag is set in the EDMA_FERR or EDMA_NERR registers, it generates an SERR, SMI, or SCI special cycle when enabled in the SERRCMD, SMICMD, or SCICMD registers, or a MCERR# on the FSB when enabled in the MCERRCMD, respectively. Note that only one message type can be enabled. All channels are expected to use the same reporting structure, so only one 8-bit register is implemented.

Table 292. Offset A0h: EDMA_SCICMD – EDMA SCI Command Register (Sheet 1 of 2)

Device: 1	**Function:** 0
Offset: A0h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07	SCI_DSCPERR	**Descriptor Address Type/Range Error SCI Enable:** Generate SCI if bit 7, 15, 23, or 31 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
06	SCI_DSCPAE	**Descriptor Address Alignment Error SCI Enable:** Generate SCI if bit 6, 14, 22, or 30 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
05	SCI_SRCERR	**Source Address Type/Range Error SCI Enable:** Generate SCI if bit 5, 13, 21, or 29 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
04	Reserved	Reserved	0b	

Table 292. Offset A0h: EDMA_SCICMD – EDMA SCI Command Register (Sheet 2 of 2)

Device: 1	**Function:** 0
Offset: A0h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
03	SCI_DESTERR	**Destination Address Type/Range Error SCI Enable:** Generate SCI if bit 3, 11, 19, or 27 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
02	Reserved	Reserved	0b	
01	SCI_MDPE1	**Memory Data Parity Error SCI Enable:** Generate SCI if bit 1, 9, 17, or 25 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
00	SCI_IWE	**Illegal Write Error SCI Enable:** Generate SCI if bit 0, 8, 16, or 24 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW

13.3.1.20 Offset A4h: EDMA_SMICMD – EDMA SMI Command Register

This register enables various errors to generate an SMI special cycle to the IICH. When an error flag is set in the EDMA_FERR or EDMA_NERR registers, it generates an SERR, SMI, or SCI special cycle when enabled in the SERRCMD, SMICMD, or SCICMD registers, or a MCERR# on the FSB when enabled in the MCERRCMD, respectively. Note that only one message type can be enabled. All channels are expected to use the same reporting structure, so only one 8-bit register is implemented.

Table 293. Offset A4h: EDMA_SMICMD – EDMA SMI Command Register (Sheet 1 of 2)

Device: 1	**Function:** 0
Offset: A4h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07	SMI_DSCPERR	**Descriptor Address Type/Range Error SMI Enable:** Generate SMI if bit 7, 15, 23, or 31 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
06	SMI_DSCPAE	**Descriptor Address Alignment Error SMI Enable:** Generate SMI if bit 6, 14, 22, or 30 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
05	SMI_SRCERR	**Source Address Type/Range Error SMI Enable:** Generate SMI if bit 5, 13, 21, or 29 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
04	Reserved	Reserved	0b	

Table 293. Offset A4h: EDMA_SMICMD – EDMA SMI Command Register (Sheet 2 of 2)

Device: 1	**Function:** 0
Offset: A4h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
03	SMI_DSTERR	**Destination Address Type/Range Error SMI Enable:** Generate SMI if bit 3, 11, 19, or 27 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
02	Reserved	Reserved	0b	
01	SMI_MDPE1	**Memory Data Parity Error SMI Enable:** Generate SMI if bit 1, 9, 17, or 25 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
00	SMI_IWE	**Illegal Write Error SMI Enable:** Generate SMI if bit 0, 8, 16, or 24 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW

13.3.1.21 Offset A8h: EDMA_SERRCMD – EDMA SERR Command Register

This register enables various errors to generate an SERR special cycle to the IICH. When an error flag is set in the EDMA_FERR or EDMA_NERR registers, it generates an SERR, SMI, or SCI special cycle when enabled in the SERRCMD, SMICMD, or SCICMD registers, or a MCERR# on the FSB when enabled in the MCERRCMD, respectively. Note that only one message type can be enabled. All channels are expected to use the same reporting structure, so only one 8-bit register is implemented.

Table 294. Offset A8h: EDMA_SERRCMD – EDMA SERR Command Register (Sheet 1 of 2)

Device: 1	**Function:** 0
Offset: A8h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07	SERR_DSCPERR	**Descriptor Address Type/Range Error SERR Enable:** Generate SERR if bit 7, 15, 23, or 31 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
06	SERR_DSCPAE	**Descriptor Address Alignment Error SERR Enable:** Generate SERR if bit 6, 14, 22, or 30 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
05	SERR_SRCERR	**Source Address Type/Range Error SERR Enable:** Generate SERR if bit 5, 13, 21, or 29 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
04	Reserved	Reserved	0b	

Table 294. Offset A8h: EDMA_SERRCMD – EDMA SERR Command Register (Sheet 2 of 2)

Device: 1	**Function:** 0
Offset: A8h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
03	SERR_DSTERR	**Destination Address Type/Range Error SERR Enable:** Generate SERR if bit 3, 11, 19, or 27 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
02	Reserved	Reserved	0b	
01	SERR_MDPE1	**Memory Data Parity Error SERR Enable:** Generate SERR if bit 1, 9, 17, or 25 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
00	SERR_IWE	**Illegal Write Error SERR Enable:** Generate SERR if bit 0, 8, 16, or 24 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW

13.3.1.22 Offset ACh: EDMA_MCERRCMD – EDMA MCERR Command Register

This register enables various errors to generate the MCERR# signal on the FSB. When an error flag is set in the EDMA_FERR or EDMA_NERR registers, it generates an SERR, SMI, or SCI special cycle when enabled in the SERRCMD, SMICMD, or SCICMD registers, or a MCERR# on the FSB when enabled in the MCERRCMD, respectively. Note that only one message type can be enabled. All channels are expected to use the same reporting structure, so only one 8-bit register is implemented.

Table 295. Offset ACh: EDMA_MCERRCMD – EDMA MCERR Command Register (Sheet 1 of 2)

Device: 1	**Function:** 0
Offset: ACh	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07	MCERR#_DSCPERR	**Descriptor Address Type/Range Error MCERR# Enable:** Generate MCERR# if bit 7, 15, 23, or 31 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
06	MCERR#_DSCPAE	**Descriptor Address Alignment Error MCERR# Enable:** Generate MCERR# if bit 6, 14, 22, or 30 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
05	MCERR#_SRCERR	**Source Address Type/Range Error MCERR# Enable:** Generate MCERR# if bit 5, 13, 21 or 29 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
04	Reserved	Reserved	0b	

Table 295. Offset ACh: EDMA_MCERRCMD – EDMA MCERR Command Register (Sheet 2 of 2)

Device: 1	**Function:** 0
Offset: ACh	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
03	MCERR#_DSTERR	**Destination Address Type/Range Error MCERR# Enable:** Generate MCERR# if bit 3, 11, 19, or 27 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
02	Reserved	Reserved	0b	
01	MCERR#_MDPE1	**Memory Data Parity Error MCERR# Enable:** Generate MCERR# if bit 1, 9, 17, or 25 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW
00	MCERR#_IWE	**Illegal Write Error MCERR# Enable:** Generate MCERR# if bit 0, 8, 16, or 24 is set in any of the EDMA_FERR or EDMA_NERR registers. 0 = Disable 1 = Enable	0b	RW

13.3.1.23 Offset B0h - B3h: MSICR – MSI Control Register

The EDMA controller generates an upstream interrupt message using Message Signaled Interrupts (MSI) to the processor, bypassing the IOxAPIC. The MSI is generated by a memory write to address 0FEEx_xxxxh. The MSI Control Register (MSICR), MSI Address Register (MSIAR) and MSI Data Register (MSIDR) support this mechanism. The default values of these registers are compatible with the default value of IOxAPIC. System software can reprogram these values, if required.

The MSI Control Register (MSICR) contains all the information related to the capability of EDMA MSI interrupts.

Table 296. Offset B0h - B3h: MSICR – MSI Control Register (Sheet 1 of 2)

Device: 1	**Function:** 0
Offset: B0 - B3h	**Size:** 32 bit
Default Value: 0002_0005h	

Bits	Name	Description	Reset Value	Access
31:24	Reserved	Reserved.	00h	
23	ADDCPBL	**Indicates 64-bit Address Capable:** Hardwired to 0 to indicate that the EDMA Controller is capable of 32-bit MSI addressing only.	0b	RO

Table 296. Offset B0h - B3h: MSICR – MSI Control Register (Sheet 2 of 2)

Device: 1	***Function:*** 0
Offset: *B0 - B3h*	***Size:*** *32 bit*
Default Value: *0002_0005h*	

Bits	Name	Description	Reset Value	Access
22:20	MME	**Multiple Message Enable:** The software writes to this field to indicate the number of allocated messages, which is aligned to a power of two. The value programmed into this field must be less than or equal to the number requested in the Multiple Messages Capable field. When MSI is enabled, the software allocates at least one message to the device. If two MSI messages are enabled, Message 0 is used for normal interrupts, and Message 1 is used for abort/error interrupts. If only one MSI message is enabled, Message 0 is used for both normal and error interrupts.	0h	RW
19:17	MMC	**Multiple Message Capable:** Hardwired to a value of 001b to indicate that the EDMA requests a capability for two messages.	001b	RO
16	MSIE	**MSI Enable:** Interrupts are generated for the conditions as described in the descriptor control register for each channel. If none of these conditions are selected, software must poll for status since no interrupts of either type are generated. 0 = Legacy interrupts are generated. 1 = MSI is generated.	0b	RW
15:08	NXT_PTR	**Next Pointer:** Pointer to the next item in the capabilities list. Hardwired to 00h to indicate that MSI is the last item in the Capabilities List.	00h	RO
07:00	CAP_ID	**Capability ID:** Hardwired to 05h to indicate that the EDMA Controller is MSI capable.	05h	RO

13.3.1.24 Offset B4h - B7h: MSIAR – MSI Address Register

The MSI Address Register (MSIAR) contains all the address related information to route MSI interrupts.

Table 297. Offset B4h - B7h: MSIAR – MSI Address Register (Sheet 1 of 2)

Device: 1	***Function:*** 0
Offset: *B4 - B7h*	***Size:*** *32 bit*
Default Value: *FEE0_0000h*	

Bits	Name	Description	Reset Value	Access
31:20	MSIADD	**Address:** Most significant 12 bits of the 32-bit address.	FEEh	RW
19:12	MSIDID	**Destination ID:** Should reflect the 63:56 bits of IOxAPIC redirection table entry.	00h	RW
11:04	MSIEDID	**Extended Destination ID:** Should reflect the 55:48 bits of IOxAPIC redirection table entry.	00h	RW

Table 297. Offset B4h - B7h: MSIAR – MSI Address Register (Sheet 2 of 2)

Device: 1	**Function:** 0
Offset: B4 - B7h	**Size:** 32 bit
Default Value: FEE0_0000h	

Bits	Name	Description	Reset Value	Access
03	RDRCTID	**Redirection Hint:** Allows the interrupt message to be redirected. 0 = Direct. Message is delivered to the agent listed in bits 19:12 1 = Redirect. Message is delivered to an agent with a lower interrupt priority. This can be derived from bits 10:08 of the Data field	0b	RW
02	DSTMD	**Destination Mode:** Used only if the Redirection Hint bit is set to 1. 0 = Physical 1 = Logical	0b	RW
01:00	Reserved	Reserved.	00b	

13.3.1.25 Offset B8h - B9h: MSIDR – MSI Data Register

The MSI Data Register (MSIDR) contains all the data-related information to route MSI interrupts.

Table 298. Offset B8h - B9h: MSIDR – MSI Data Register (Sheet 1 of 2)

Device: 1	**Function:** 0
Offset: B8 - B9h	**Size:** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
15	TRGMD	**Trigger Mode:** Software must set this to be the same as the corresponding bit in the I/O Redirection Table for that interrupt. 0 = Edge 1 = Level	0b	RW
14	DLVSTS	**Delivery Status:** If using edge-triggered interrupts, this is always 1, since only the assertion is sent. If using level-triggered interrupts, then this bit indicates the state of the interrupt input.	0b	RW
13:12	Reserved	Reserved	00b	

Table 298. Offset B8h - B9h: MSIDR – MSI Data Register (Sheet 2 of 2)

Device: 1	***Function:*** 0
Offset: *B8 - B9h*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
11	DSTNMD	**Destination Mode:** Software must set this to be the same as bit 2 of MSIAR. 0 = Physical 1 = Logical.	0b	RW
10:08	DLVMD	**Delivery Mode:** Software must set this to be the same as the corresponding bits in the I/O Redirection Table for that interrupt. 000 Fixed 001 Lowest Priority 010 SMI/PMI 011 Reserved 100 NMI 101 INIT 110 Reserved 111 ExtINT	0h	RW
07:00	INTRPTV	**Interrupt Vector:** Software must set this to be the same as the corresponding bits in the I/O Redirection Table for that interrupt.	00h	RW

13.4 Device 2, Function 0: PCI Express* Port A Standard and Enhanced Registers

Device 2 is the PCI Express Port A (in x8 mode) or port A0 (in x4 mode) virtual PCI-to-PCI bridge. The registers described here include both the standard configuration space and the enhanced configuration space (starting at offset 100h).

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 299. PCI Express Port A Standard and Enhanced Configuration Register Map (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	01h	VID	Vendor Identification Register	8086h	RO
02h	03h	DID	Device Identification Register	35B6h	RO
04h	05h	PCICMD	PCI Command Register	0000h	RO, RW
06h	07h	PCISTS	PCI Status Register	0010h	RO, RWC
08h	08h	RID	Revision Identification Register	00h	RO
0Ah	0Ah	SUBC	Sub-Class Code Register	04h	RO
0Bh	0Bh	BCC	Base Class Code Register	06h	RO
0Ch	0Ch	CLS	Cache Line Size Register	00h	RW
0Eh	0Eh	HDR	Header Type Register	01h	RO
18h	18h	PBUSN	Primary Bus Number Register	00h	RO
19h	19h	SBUSN	Secondary Bus Number Register	00h	RW
1Ah	1Ah	SUBUSN	Subordinate Bus Number Register	00h	RW
1Ch	1Ch	IOBASE	I/O Base Address Register	F0h	RO, RW
1Dh	1Dh	IOLIMIT	I/O Limit Address Register	00h	RW
1E	1Fh	SECSTS	Secondary Status Register	0000h	RO, RWC
20h	21h	MBASE	Memory Base Address Register	FFF0h	RO, RW
22h	23h	MLIMIT	Memory Limit Address Register	0000h	RO, RW
24h	25h	PMBASE	Prefetchable Memory Base Address Register	FFF1h	RO, RW
26h	27h	PMLIMIT	Prefetchable Memory Limit Address Register	0001h	RO, RW
28h	28h	PMBASU	Prefetchable Memory Base Upper Address Register	0Fh	RO, RW
2Ch	2Ch	PMLMTU	Prefetchable Memory Limit Upper Address Register	00h	RO, RW
34h	34h	CAPPTR	Capabilities Pointer Register	50h	RO
3Ch	3Ch	INTRLINE	Interrupt Line Register	00h	RW
3Dh	3Dh	INTRPIN	Interrupt Pin Register	01h	RWO
3Eh	3Eh	BCTRL	Bridge Control Register	00h	RO, RW
44h	44h	VSCMD0	Vendor-Specific Command Byte 0 Register	00h	RW
45h	45h	VSCMD1	Vendor-Specific Command Byte 1 Register	00h	RO, RW, RWS
46h	46h	VSSTS0	Vendor-Specific Status Byte 0 Register	00h	RO
47h	47h	VSSTS1	Vendor-Specific Status Byte 1 Register	00h	RO, RWC
50h	50h	PMCAPID	Power Management Capabilities Structure Register	01h	RO
51h	51h	PMNPTR	Power Management Next Capabilities Pointer Register	58h	RO
52h	53h	PMCAPA	Power Management Capabilities Register	C822h	RO
54h	55h	PMCSR	Power Management Status and Control Register	0000h	RO, RW
56h	56h	PMCSRBSE	Power Management Status and Control Bridge Extensions Register	00h	RO
58h	58h	MSICAPID	MSI Capabilities Structure Register	05h	RO

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 299. PCI Express Port A Standard and Enhanced Configuration Register Map (Sheet 2 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
59h	59h	MSINPTR	MSI Next Capabilities Pointer Register	64h	RO
5Ah	5Bh	MSICAPA	MSI Capabilities Register	0002h	RO, RW
5Ch	5Fh	MSIAR	MSI Address for PCI Express Register	FEE0_0000h	RW
60h	61h	MSIDR	MSI Data Register	0000h	RW
64h	64h	PEACAPID	PCI Express Features Capabilities ID Register	10h	RO
65h	65h	PEANPTR	PCI Express Next Capabilities Pointer Register	00h	RO
66h	67h	PEACAPA	PCI Express Features Capabilities Register	0041h	RO
68h	6Bh	PEADEVCAP	PCI Express Device Capabilities Register	0000_0001h	RO
6Ch	6Dh	PEADEVCTL	PCI Express Device Control Register	0000h	RO, RW
6Eh	6Fh	PEADEVSTS	PCI Express Device Status Register	0000h	RO, RWC
70h	73h	PEALNKCAP	PCI Express Link Capabilities Register	0203_E481h	RO, RWO
74h	75h	PEALNKCTL	PCI Express Link Control Register	0000h	RO, RW, WO
76h	77h	PEALNKSTS	PCI Express Link Status Register	1001h	RO, RWO
78h	7Bh	PEASLTCAP	PCI Express Slot Capabilities Register	0000_0000h	RO, RWO
7Ch	7Dh	PEASLTCTL	PCI Express Slot Control Register	01C0h	RO, RW
7Eh	7Fh	PEASLTSTS	PCI Express Slot Status Register	0040h	RO, RWC
80h	83h	PEARPCTL	PCI Express Root Port Control Register	0000_0000h	RO, RW
84h	87h	PEARPSTS	PCI Express Root Port Status Register	0000_0000h	RO, RWC
100h	103h	ENHCAPST	Enhanced Capability Structure Register	0001_0001h	RO
104h	107h	UNCERRSTS	Uncorrectable Error Status Register	0000_0000h	RO, RWC
108h	10Bh	UNCERRMSK	Uncorrectable Error Mask Register	0000_0000h	RO, RW
10Ch	10Fh	UNCERRSEV	Uncorrectable Error Severity Register	0006_2010h	RO, RW
110h	113h	CORERRSTS	Correctable Error Status Register	0000_0000h	RO, RWC
114h	117h	CORERRMSK	Correctable Error Mask Register	0000_0000h	RO, RW
118h	11Bh	AERCACR	Advanced Error Capabilities and Control Register	000_00A0h	RO
11Ch	11Fh	HDRLOG0	Header Log DW 0 (1st 32 bits) Register	0000_0000h	RO
120h	123h	HDRLOG1	Header Log DW 1 (2nd 32 bits) Register	0000_0000h	RO
124h	127h	HDRLOG2	Header Log DW 2 (3rd 32 bits) Register	0000_0000h	RO
128h	12Bh	HDRLOG3	Header Log DW 3 (4th 32 bits) Register	0000_0000h	RO
12Ch	12Fh	RPERRCMD	Root (Port) Error Command Register	0000_0000h	RO, RW
130h	133h	RPERRMSTS	Root (Port) Error Message Status Register	0000_0000h	RO, RWC
134h	137h	ERRSID	Error Source ID Register	0000_0000h	RO
140h	143h	PEAUNITERR	PCI Express Unit Error Status Register	0000_0000h	RO, RWC
144h	147h	PEAMASKERR	PCI Express Unit Mask Error Register	0000_E000h	RO, RW
148h	14Bh	PEAERRDOCMD	PCI Express Error Do Command Register	0000_0000h	RO, RW
14Ch	14Fh	UNCEDMASK	Uncorrectable Error Detect Mask Register	0000_0000h	RO, RW
150h	153h	COREDMASK	Correctable Error Detect Mask Register	0000_0000h	RO, RW
158h	15Bh	PEAUNITDMASK	PCI Express Unit Error Detect Mask Register	0000_0000h	RO, RW
160h	163h	PEAFERR	PCI Express First Error Register	0000_0000h	RO, RWC
164h	167h	PEANERR	PCI Express Next Error Register	0000_0000h	RO, RWC
168h	16Bh	PEAERRCTL	PCI Express Port A Error Control Register	0000_0000h	RO, RW, RWS

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

13.4.1 Register Details

13.4.1.1 Offset 00 - 01h: VID – Vendor Identification Register

The VID Register contains the vendor identification number. This 16-bit register combined with the Device Identification Register uniquely identify any PCI device.

Table 300. Offset 00 - 01h: VID – Vendor Identification Register

Device: 2	***Function:*** 0
Offset: *00 - 01h*	***Size:*** *16 bit*
Default Value: *8086h*	

Bits	Name	Description	Reset Value	Access
15:00	VID	**Vendor Identification Device:** This is a 16-bit value assigned to Intel.	8086h	RO

13.4.1.2 Offset 02 - 03h: DID – Device Identification Register

This 16-bit register combined with the Vendor Identification register uniquely identifies any PCI device. Note that the Device ID changes for each of the PCI Express ports, starting with 35B6h for Device 2.

Table 301. Offset 02 - 03h: DID – Device Identification Register

Device: 2	***Function:*** 0
Offset: *02 - 03h*	***Size:*** *16 bit*
Default Value: *35B6h*	

Bits	Name	Description	Reset Value	Access
15:00	DID	**Device Identification Number:** This is a 16-bit value assigned to the IMCH Device 2, Function 0.	35B6h	RO

13.4.1.3 Offset 04 - 05h: PCICMD – PCI Command Register

Many of these bits are not applicable since the primary side of this device is not an actual PCI bus.

Table 302. Offset 04 - 05h: PCICMD – PCI Command Register (Sheet 1 of 2)

Device: 2	***Function:*** 0
Offset: *04 - 05h*	***Size:*** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
15:11	Reserved	Reserved.	00h	
10	INTXD	**INTx Assertion Disable:** Controls the ability of thePCI Express device to assert INTx interrupts. When set, devices are prevented from asserting INTx. This bit only applies to legacy interrupts and not MSIs. Also, this bit has no affect on PCI Express messages that are converted to legacy interrupts. These are only internal, device generated interrupts. 0 = Enable INTx assertion 1 = Disable INTx assertion	0b	RW

Table 302. Offset 04 - 05h: PCICMD – PCI Command Register (Sheet 2 of 2)

Device: 2	***Function:*** 0
Offset: *04 - 05h*	***Size:*** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
09	FB2B	**Fast Back-to-Back Enable:** Not Applicable-hardwired to 0.	0b	RO
08	SERRE	**SERR Enable:** This bit is a global enable bit for Device SERR messaging. The IMCH does not have an SERR# signal. The IMCH communicates the SERR# condition by sending an SERR message to the IICH via NSI. 0 = No SERR message is generated by the IMCH for Device 2 (unless enabled through enhanced configuration registers). 1 = Enable SERR, SCI, or SMI messages or asserting MCERR# for specific Device 2 error conditions.	0b	RW
07	ADSTEP	**Address/Data Stepping:** Not applicable.	0b	RO
06	PERRE	**Parity Error Enable:** This bit determines the device behavior on detection of a parity error. See the *PCI Express* Interface Specification, Rev 1.0a*, for details. 0 = Parity Errors are logged in the status register, but no other action is taken. 1 = Normal action is taken upon detection of Parity Error, as well as logging.	0b	RW
05	VPS	**VGA palette snoop:** Not applicable.	0b	RO
04	MWIE	**Memory Write and Invalidate Enable:** Not applicable.	0b	RO
03	SCE	**Special Cycle Enable:** Not applicable.	0b	RO
02	BME	**Bus Master Enable:** This bit controls the PCI Express port's ability to issue memory and I/O read/write requests on behalf of subordinate devices. MSI interrupt messages are in-band memory writes, and clearing this bit disables MSI interrupt messages. 0 = Disable. The port does not respond to any I/O or memory transaction originating on the secondary interface. 1 = Enable.	0b	RW
01	MAE	**Memory Access Enable:** Controls access to the Memory and Prefetchable memory address ranges. 0 = Disable all of device memory space 1 = Enable	0b	RW
00	IOAE	**I/O Access Enable:** Controls access to the I/O address range defined in the IOBASE and IOLIMIT registers. 0 = Disable device I/O space 1 = Enable	0b	RW

13.4.1.4 Offset 06 - 07h: PCISTS – PCI Status Register

PCISTS is a 16-bit status register that reports the occurrence of error conditions associated with the primary side of the "virtual" PCI-PCI bridge embedded within the IMCH.

Table 303. Offset 06 - 07h: PCISTS – PCI Status Register (Sheet 1 of 2)

Device: 2 ***Function:*** 0
Offset: *06 - 07h* ***Size:*** *16 bit*
Default Value: *0010h*

Bits	Name	Description	Reset Value	Access
15	DPE	**Detected Parity Error:** Parity is supported on the primary side of this device. Since the parity is not checked on the downstream side from the core, this bit can never be set. 0 = No Parity Error detected	0b	RO
14	SSE	**Signaled System Error:** This bit indicates whether or not a NSI SERR message was generated by this device. Errors can be generated locally or remotely. The remote errors are received through virtual messages that are forwarded for reporting. 0 = SERR message not generated by this device (remote). 1 = This device was the source of fatal or non-fatal error that has been enabled for generation of a System Error (local). Software clears this bit by writing a '1' to the bit location.	0b	RWC
13	RMAS	**Received Master Abort Status:** Indicates whether or not this PCI Express device received a completion with Unsupported Request Completion status. 0 = No Master Abort received. 1 = Set when this PCI Express device receives a completion with Unsupported Request Completion Status. Software clears this bit by writing a '1' to the bit location.	0b	RWC
12	RTAS	**Received Target Abort Status:** Indicates whether or not this PCI Express device received a completion with Completer Abort Completion Status. 0 = No Target Abort received. 1 = Set when this PCI Express device receives a completion with Completer Abort Completion Status. Software clears this bit by writing a '1' to the bit location.	0b	RWC
11	STAS	**Signaled Target Abort Status:** Not applicable to the primary side. 0 = This PCI Express device has not completed a request using Completer Abort Completion Status. 1 = This PCI Express device completed a request using Completer Abort Completion Status.	0b	RO
10:09	DEVT	**DEVSEL# Timing:** Not Applicable. Hardwired to 0.	00b	RO
08	DPD	**Master Data Parity Error Detected:** Parity is supported on the primary side of this device. 0 = No Master Parity Error detected. 1 = Set when this PCI Express device receives a completion marked poisoned, or when this device poisons a write Request. This bit can only be set if the Parity Error Enable bit is set. Software clears this bit by writing a '1' to the bit location.	0b	RWC
07	FB2B	**Fast Back-to-Back:** Not Applicable	0b	RO
06	Reserved	Reserved	0b	
05	C66M	**Capable 66MHz:** Not Applicable	0b	RO
04	CAPL	**Capabilities List:** Hardwired to 1 to indicate the presence of an Extended Capability List item.	1b	RO

Table 303. Offset 06 - 07h: PCISTS – PCI Status Register (Sheet 2 of 2)

Device: 2	***Function:*** 0
Offset: *06 - 07h*	***Size:*** *16 bit*
Default Value: *0010h*	

Bits	Name	Description	Reset Value	Access
03	INTXS	**INTx Status:** This bit does not get set for interrupts forwarded up from downstream devices, or for messages converted to interrupts by the root port. The INTx Assertion Disable bit has no effect on the setting of this bit. This bit is not set for an MSI. 0 = An INTx interrupt is not pending internal to this device. 1 = An INTx interrupt is pending internal to this device.	0b	RO
02:00	Reserved	Reserved	00h	

13.4.1.5 Offset 08h: RID – Revision Identification Register

This register contains the revision number of the IMCH Device 2 device.

Table 304. Offset 08h: RID – Revision Identification Register

Device: 2	***Function:*** 0
Offset: *08h*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:00	RID	**Revision Identification Number:** This value indicates the revision identification number for the device 2. It is always the same as the value in Device 0 RID. 00h = A-0 stepping	00h	RO

13.4.1.6 Offset 0Ah: SUBC – Sub-Class Code Register

This register contains the Sub-Class Code for the device.

Table 305. Offset 0Ah: SUBC – Sub-Class Code Register

Device: 2	***Function:*** 0
Offset: *0Ah*	***Size:*** *8 bit*
Default Value: *04h*	

Bits	Name	Description	Reset Value	Access
07:00	SUBC	**Sub-Class Code:** This value indicates the category of Bridge into which device falls.04h = PCI to PCI Bridge.	04h	RO

13.4.1.7 Offset 0Bh: BCC – Base Class Code Register

This register contains the Base Class Code of the IMCH device 2.

Table 306. Offset 0Bh: BCC – Base Class Code Register

Device: 2	***Function:*** 0
Offset: *0Bh*	***Size:*** *8 bit*
Default Value: *06h*	

Bits	Name	Description	Reset Value	Access
07:00	BASEC	**Base Class Code:** This value indicates the Base Class Code for the device. 06h = Bridge device	06h	RO

13.4.1.8 Offset 0Ch: CLS – Cache Line Size Register

This register is normally set by system firmware and OS to the system cache line size. Legacy PCI 2.3 software may not always be able to program this field correctly, especially in the case of hot plug devices. It is implemented as a read-write field for legacy compatibility purposes, but has no effect on this device's functionality.

Table 307. Offset 0Ch: CLS – Cache Line Size Register

Device: 2	***Function:*** 0
Offset: *0Ch*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:00	CLS	**Cache Line Size:** This register is set by BIOS or OS to the system cache line size. Implemented as read-write field only for compatibility reasons. It has no effect on the device's functionality.	00h	RW

13.4.1.9 Offset 0Eh: HDR – Header Type Register

This register identifies the header layout of the configuration space.

Table 308. Offset 0Eh: HDR – Header Type Register

Device: 2	***Function:*** 0
Offset: *0Eh*	***Size:*** *8 bit*
Default Value: *01h*	

Bits	Name	Description	Reset Value	Access
07:00	HDR	**Header Type Register:** This value indicates the Header Type of the device. 01h = single-function device with Bridge layout.	01h	RO

13.4.1.10 Offset 18h: PBUSN – Primary Bus Number Register

This register identifies that "virtual" PCI-to-PCI bridge is connected to bus 0.

Table 309. Offset 18h: PBUSN – Primary Bus Number Register

Device: 2	***Function:*** 0
Offset: *18h*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:00	BUSN	**Primary Bus Number:** Configuration software typically programs this field with the number of the bus on the primary side of the bridge. Since Device 2 is an internal device and its primary bus is always 0, these bits are hardwired to 0.	00h	RO

13.4.1.11 Offset 19h: SBUSN – Secondary Bus Number Register

This register identifies the bus number assigned to the second bus side of the "virtual" PCI-to-PCI bridge (the PCI Express connection). This number is programmed by the PCI configuration software to allow mapping of configuration cycles to a secondary bridge device connected to PCI Express.

Table 310. Offset 19h: SBUSN – Secondary Bus Number Register

Device: 2	***Function:*** 0
Offset: *19h*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:00	BUSN	**Secondary Bus Number:** This field is programmed by configuration software with the lowest bus number of the PCI Express port.	00h	RW

13.4.1.12 Offset 1Ah: SUBUSN – Subordinate Bus Number Register

This register is programmed by PCI configuration software to the highest numbered subordinate bus (if any) that resides below another bridge device below the secondary PCI Express interface.

Table 311. Offset 1Ah: SUBUSN – Subordinate Bus Number Register

Device: 2	***Function:*** 0
Offset: *1Ah*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:00	BUSN	**Subordinate Bus Number:** This register is programmed by configuration software with the number of the highest subordinate bus that lies behind the device bridge.	00h	RW

13.4.1.13 Offset 1Ch: IOBASE – I/O Base Address Register

The IOBASE and IOLIMIT registers control the processor-to-PCI Express I/O access routing based on the following formula:

IO_BASE ≤ address ≤ IO_LIMIT

Only the upper four bits are programmable. For the purpose of address decode address bits A[11:00] are treated as 0. Thus the bottom of the defined I/O address range is aligned to a 4 Kbyte boundary.

Table 312. Offset 1Ch: IOBASE – I/O Base Address Register

Device: 2	***Function:*** 0
Offset: *1Ch*	***Size:*** *8 bit*
Default Value: *F0h*	

Bits	Name	Description	Reset Value	Access
07:04	IOBASE	**I/O Address Base:** Corresponds to A[15:12] of the I/O addresses passed by the device bridge to PCI Express.	Fh	RW
03:00	IOBM	**I/O Addressing mode:** These bits are hardwired to 0. 0h = 16-bit I/O addressing All other bit combinations are not supported.	0h	RO

13.4.1.14 Offset 1Dh: IOLIMIT – I/O Limit Address Register

This register controls the processor to PCI Express I/O access routing based on the following formula:

IO_BASE ≤ address ≤ IO_LIMIT

Only the upper four bits are programmable. For the purpose of address decode address bits A[11:00] are assumed to be FFFh. Thus, the top of the defined I/O address range is at the top of a 4 Kbyte aligned address block.

Table 313. Offset 1Dh: IOLIMIT – I/O Limit Address Register

Device: 2	***Function:*** 0
Offset: *1Dh*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:04	IOLIMIT	**I/O Address Limit:** Corresponds to A[15:12] of the I/O address limit of device. Devices between this upper limit and IOBASE2 are passed to PCI Express.	0h	RW
03:00	IOLM	**I/O Addressing mode:** These bits are hardwired to 0. 0h = 16-bit I/O addressing All other bit combinations are not supported.	0h	RO

13.4.1.15 Offset 1E - 1Fh: SECSTS – Secondary Status Register

SECSTS is a 16-bit status register that reports the occurrence of error conditions associated with the secondary side (e.g., PCI Express side) of the "virtual" PCI-PCI bridge embedded within the Intel® 3100 Chipset.

Table 314. Offset 1E - 1Fh: SECSTS – Secondary Status Register

Device: 2	***Function:*** 0
Offset: *1E - 1Fh*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15	2DPE	**Detected Parity Error:** This bit is set by the PCI Express Port logic when the secondary side receives a poisoned TLP, regardless of the state of the Parity Error Enable bit. Software clears this bit by writing a '1' to the bit location. See the *PCI Express* Interface Specification, Rev 1.0a* for details. 0 = No parity Error detected. 1 = Parity Error Detected (poisoned TLP received).	0b	RWC
14	2RSE	**Received System Error:** Indicates whether or not an ERR_FATAL or ERR_NONFATAL message was received via PCI Express. 0 = Error message not received by this device. 1 = This device received fatal or non-fatal error message via PCI Express. Software clears this bit by writing a '1' to the bit location. This bit is not set for virtual messages.	0b	RWC
13	2RMAS	**Received Master Abort Status:** Indicates whether or not this PCI Express device received a completion with Unsupported Request Completion status. 0 = No Master Abort received. Software clears this bit by writing a '1' to the bit location. 1 = Set when this PCI Express device receives a completion with Unsupported Request Completion Status.	0b	RWC
12	2RTAS	**Received Target Abort Status:** Indicates whether or not this PCI Express device received a completion with Completer Abort Completion Status. 0 = No Target Abort received. Software clears this bit by writing a '1' to the bit location. 1 = Set when this PCI Express device receives a completion with Completer Abort Completion Status.	0b	RWC
11	STAS	**Signaled Target Abort Status:** Indicates whether or not this PCI Express device completed a request using Completer Abort Completion Status. 0 = No Target Abort signaled. Software clears this bit by writing a '1' to the bit location. 1 = Set when this PCI Express device completes a request using Completer Abort Completion Status.	0b	RWC
10:09	DEVT	**DEVSEL# Timing:** Not Applicable	00b	RO
08	DPD	**Master Data Parity Error Detected:** Parity is supported on the secondary side of this device. 0 = No Master Parity Error detected. Software clears this bit by writing a '1' to the bit location. 1 = Set when this PCI Express device receives a completion marked poisoned, or when this device poisons a write Request. This bit can only be set if the Parity Error Enable bit is set.	0b	RWC
07:00	Reserved	Reserved	0b	

13.4.1.16 Offset 20 - 21h: MBASE – Memory Base Address Register

This register controls the processor to PCI Express non-prefetchable memory access routing based on the following formula:

MEMORY_BASE ≤ address ≤ MEMORY_LIMIT

Note: The upper 12 bits of the register are read/write and correspond to the upper 12 address bits A[31:20] of the 32-bit address. The bottom four bits of this register are read-only and return zeroes when read. This register must be initialized by the configuration software. For the purpose of address decode address bits A[19:00] of the Memory Base Address are assumed to be 0. Similarly, the bridge assumes that the lower 20 bits of the Memory Limit Address (A[19:00]) are F_FFFFh. Thus, the bottom of the defined memory address range are aligned to a 1 Mbyte boundary, and the top of the defined memory range is at the top of a 1 Mbyte memory block. Memory ranges covered by MBASE and MLIMIT registers are used to map non-prefetchable PCI Express address ranges (typically where control/status memory-mapped I/O data structures of the graphics controller resides) and PMBASE and PMLIMIT are used to map prefetchable address ranges (typically graphics local memory). This segregation allows application of USWC space attribute to be performed in a true plug-and-play manner to the prefetchable address range for improved PCI Express memory access performance.

Configuration software is responsible for programming all address range registers (prefetchable, non-prefetchable) with the values that provide exclusive address ranges (for example, to prevent overlap with each other and/or with the ranges covered with the main memory). There is no provision in the Intel® 3100 Chipset hardware to enforce prevention of overlap and operations of the system in the case of overlap are not guaranteed.

Table 315. Offset 20 - 21h: MBASE – Memory Base Address Register

Device: *2* ***Function:*** *0*
Offset: *20 - 21h* ***Size:*** *16 bit*
Default Value: *FFF0h*

Bits	Name	Description	Reset Value	Access
15:04	MBASE	**Memory Address Base:** Corresponds to A[31:20] of the lower limit of the memory range that are passed by the Device 2 bridge to PCI Express.	FFFh	RW
03:00	Reserved	Reserved	0h	

13.4.1.17 Offset 22 - 23h: MLIMIT – Memory Limit Address Register

This register controls the processor to PCI Express non-prefetchable memory access routing based on the following formula:

MEMORY_BASE ≤ address ≤ MEMORY_LIMIT

The upper 12 bits of the register are read/write and correspond to the upper 12 address bits A[31:20] of the 32-bit address. The bottom four bits of this register are read-only and return zeroes when read. This register must be initialized by configuration software. For the purpose of address decode address bits A[19:00] are assumed to be FFFFFh. Thus, the top of the defined memory address range is at the top of a 1 Mbyte aligned memory block.

Note: Memory ranges covered by MBASE and MLIMIT registers are used to map non-prefetchable PCI Express address ranges (typically where control/status memory-

mapped I/O data structures of the graphics controller reside) and PMBASE and PMLIMIT are used to map prefetchable address ranges (typically graphics local memory). This segregation allows application of USWC space attribute to be performed in a true plug-and-play manner to the prefetchable address range for improved PCI Express memory access performance.

Configuration software is responsible for programming all address range registers (prefetchable, non-prefetchable) with the values that provide exclusive address ranges; for example, to prevent overlap with each other and/or with the ranges covered with the main memory. There is no provision in the Intel® 3100 Chipset hardware to enforce prevention of overlap and operations of the system in the case of overlap are not guaranteed.

Table 316. Offset 22 - 23h: MLIMIT – Memory Limit Address Register

Device: *2* ***Function:*** *0*
Offset: *22 - 23h* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:04	MLIMIT	**Memory Address Limit:** Corresponds to A[31:20] of the memory address that corresponds to the upper limit of the range of memory accesses that are passed by the device bridge to PCI Express.	000h	RW
03:00	Reserved	Reserved	0h	

13.4.1.18 Offset 24 - 25h: PMBASE – Prefetchable Memory Base Address Register

This PMBASE and PMLIMIT register controls the processor to PCI Express prefetchable memory accesses. The upper 12 bits of the register are read/write and correspond to the upper 12 address bits A[31:20] of the 32-bit address. For the purpose of address decode, bits A[19:00] of the Prefetchable Memory Base Address are assumed to be 0. Similarly, the bridge assumes that the lower 20 bits of the Prefetchable Memory Limit Address (A[19:00]) are FFFFFh. Thus, the bottom of the defined memory address range are aligned to a 1 Mbyte boundary, and the top of the defined memory range are at the top of a 1 Mbyte memory block.

The bottom 4 bits of both the Prefetchable Memory Base and Prefetchable Memory Limit registers are read-only, contain the same value, and encode whether or not the bridge supports 64-bit addresses. If these four bits have the value 0h, then the bridge supports only 32 bit addresses. If these four bits have the value 01h, then the bridge supports 64-bit addresses and the Prefetchable Base Upper 32 Bits and Prefetchable Limit Upper 32 Bits registers hold the rest of the 64-bit prefetchable base and limit addresses respectively. All other encodings are reserved.

Table 317. Offset 24 - 25h: PMBASE – Prefetchable Memory Base Address Register

Device: 2	**Function:** 0
Offset: 24 - 25h	**Size:** 16 bit
Default Value: FFF1h	

Bits	Name	Description	Reset Value	Access
15:04	PMBASE	**Prefetchable Memory Address Base:** Corresponds to A[31:20] of the lower limit of the address range passed by bridge device across PCI Express.	FFFh	RW
03:01	MAMB	**Memory Addressing Mode.** These bits are read-only with a value of zero, all other values are reserved.	0h	RO
00	MBUAE	**Memory Base Upper Address Enabled:** 0 = Disabled 1 = Enabled - Indicates that the base address is further defined by the upper address bits of the memory base upper address register.	1h	RO

13.4.1.19 Offset 26 - 27h: PMLIMIT – Prefetchable Memory Limit Address Register

This register controls the processor to PCI Express prefetchable memory accesses. The upper 12 bits of the register are read/write and correspond to the upper 12 address bits A[31:20] of the 32-bit address. For the purpose of address decode, bits A[19:00] are assumed to be FFFFFh. Thus, the top of the defined memory address range are at the top of a 1 Mbyte aligned memory block.

Table 318. Offset 26 - 27h: PMLIMIT – Prefetchable Memory Limit Address Register

Device: 2	**Function:** 0
Offset: 26 - 27h	**Size:** 16 bit
Default Value: 0001h	

Bits	Name	Description	Reset Value	Access
15:04	PMLIMIT	**Prefetchable Memory Address Limit:** Corresponds to A[31:20] of the upper limit of the address range passed by bridge Device 2 across PCI Express.	000h	RW
03:01	MAML	**Memory Addressing Mode.** These bits are read-only with a value of zero, all other values are reserved.	0h	RO
00	MLUAE	**Memory Limit Upper Address Enabled:** 0 = Disabled 1 = Enabled - Indicates that the limit address is further expanded/defined by the upper address bits of the memory limit upper address register.	1h	RO

13.4.1.20 Offset 28h: PMBASU – Prefetchable Memory Base Upper Address Register

This register expands the prefetchable memory base address by four bits. All other bits are reserved.

Table 319. Offset 28h: PMBASU – Prefetchable Memory Base Upper Address Register

Device: *2*	***Function:*** *0*
Offset: *28h*	***Size:*** *8 bit*
Default Value: *0Fh*	

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	0h	
03:00	BUA	**Base Upper Address bits:** These four bits expands the prefetchable address base to 36 bits. Corresponds to A[35:32] of the lower limit of the address range passed by bridge device across the PCI Express interface.	Fh	RW

13.4.1.21 Offset 2Ch: PMLMTU – Prefetchable Memory Limit Upper Address Register

This register expands the prefetchable memory limit address by four bits. All other bits are reserved.

Table 320. Offset 2Ch: PMLMTU – Prefetchable Memory Limit Upper Address Register

Device: *2*	***Function:*** *0*
Offset: *2Ch*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	0h	
03:00	LUA	**Limit Upper Address bits:** These four bits expands the prefetchable address limit to 36 bits. Corresponds to A[35:32] of the upper limit of the address range passed by bridge device across the PCI Express interface.	0h	RW

13.4.1.22 Offset 34h: CAPPTR – Capabilities Pointer Register

The CAPPTR provides the offset that is the pointer to the location where the first set of capabilities registers is located.

Table 321. Offset 34h: CAPPTR – Capabilities Pointer Register

Device: *2*	***Function:*** *0*
Offset: *34h*	***Size:*** *8 bit*
Default Value: *50h*	

Bits	Name	Description	Reset Value	Access
07:00	CAP_PTR	**Capabilities Pointer:** Pointer to first PCI Express Capabilities Structure register block, which is the first of the chain of capabilities.	50h	RO

13.4.1.23 Offset 3Ch: INTRLINE – Interrupt Line Register

Table 322. Offset 3Ch: INTRLINE – Interrupt Line Register

Device: 2	***Function:*** 0
Offset: *3Ch*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:00	INTRC	**Interrupt Connection:** BIOS writes the interrupt routing information to this register to indicate which input of the interrupt controller connects to this device.	00h	RW

13.4.1.24 Offset 3Dh: INTRPIN – Interrupt Pin Register

Table 323. Offset 3Dh: INTRPIN – Interrupt Pin Register

Device: 2	***Function:*** 0
Offset: *3Dh*	***Size:*** *8 bit*
Default Value: *01h*	

Bits	Name	Description	Reset Value	Access
07:00	INTRP	**Interrupt Pin:** Set to '01h' to indicate PCI Express always uses INTA# as its interrupt pin. Once this register is written, the register value locks and cannot be further updated.	01h	RWO

13.4.1.25 Offset 3Eh: BCTRL – Bridge Control Register

This register provides extensions to the PCICMD register that are specific to PCI-PCI bridges. The BCTRL provides additional control for the secondary interface (e.g. PCI Express) and some bits that affect the overall behavior of the "virtual" PCI-PCI bridge embedded within the Intel® 3100 Chipset, e.g. VGA compatible address range mapping.

Table 324. Offset 3Eh: BCTRL – Bridge Control Register (Sheet 1 of 2)

Device: 2	***Function:*** 0
Offset: *3Eh*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07	Reserved	Reserved	0b	

Table 324. Offset 3Eh: BCTRL – Bridge Control Register (Sheet 2 of 2)

Device: 2	**Function:** 0
Offset: 3Eh	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
06	SRESET	**Secondary Bus Reset:** 0 = No hot reset is triggered on the link for the corresponding PCI Express port and the PCI Express hierarchy domain subordinate to the port. 1 = Setting this bit triggers a hot reset on the link for the corresponding PCI Express port and the PCI Express hierarchy domain subordinate to the port. This sends the LTSSM into the Training (or Link) Control Reset state, which necessarily implies a reset to the downstream device and all subordinate devices. Once this bit has been cleared, and the minimum transmission requirement has been met, the detect state is entered by both ends of the link. Note also that a secondary bus reset does not in general reset the primary side configuration registers of the targeted PCI Express port. This is necessary to allow software to specify special training configuration, such as entry into loopback mode.	0b	RW
05:04	Reserved	Reserved	0b	
03	VGAEN	**VGA Enable:** Controls the routing of processor initiated transactions targeting VGA compatible I/O and memory address ranges. 0 = Disable 1 = Enable **NOTE:** Only one of Device 2–3's VGAEN bits are allowed to be set. This must be enforced via software.	0b	RW
02	ISAEN	**ISA Enable:** Modifies the response by the IMCH to an I/O access issued by the processor that targets ISA I/O addresses. This applies only to I/O addresses that are enabled by the IOBASE and IOLIMIT registers. 0 = All addresses defined by the IOBASE and IOLIMIT for processor I/O transactions are mapped to PCI Express. 1 = IMCH does not forward to PCI Express any I/O transactions addressing the last 768 bytes in each 1 Kbyte block even if the addresses are within the range defined by the IOBASE and IOLIMIT registers. Instead, these cycles are forwarded to NSI where they can be subtractively or positively claimed by the ISA bridge.	0b	RW
01	2SERRE	**SERR Enable:** This bit enables or disables forwarding of ERR_COR, ERR_NONFATAL, and ERR_FATAL messages from PCI Express to NSI, where they can be converted into interrupts that are eventually delivered to the processor. 0 = Disable 1 = Enable	0b	RW
00	2PERRE	**Parity Error Response Enable:** Controls response to poisoned TLPs on PCI Express. 0 = Disable 1 = Enable	0b	RW

13.4.1.26 Offset 44h: VSCMD0 – Vendor Specific Command Byte 0 Register

This register is for vendor specific commands.

Note: It appears as a reserved register, except for Device 2 which implements this as a hot plug specific register.

Table 325. Offset 44h: VSCMD0 – Vendor Specific Command Byte 0 Register

Device: 2	***Function:*** 0
Offset: *44h*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07	PDPI	**Presence Detect Polarity Invert:** 0 = Presence Detect normal polarity. 1 = The polarity of the Presence Detect input received from the IOX is inverted.	0b	RW
06	APPI	**Attention Pushbutton Polarity Invert:** 0 = Attention Pushbutton normal polarity. 1 = The polarity of the Attention Pushbutton input received from the IOX is inverted.	0b	RW
05	PFPI	**Power Fault Polarity Invert:** 0 = Power Fault normal polarity. 1 = The polarity of the Power Fault input received from the IOX is inverted.	0b	RW
04	MPI	**MRL Polarity Invert:** 0 = MRL normal polarity. 1 = The polarity of the MRL input received from the IOX is inverted.	0b	RW
03	PCPI	**Power Control Polarity Invert:** 0 = Power Control normal polarity. 1 = The polarity of the Power Control input received from the IOX is inverted.	0b	RW
02	IPI	**Indicator Polarity Invert:** 0 = Attention and Power indicators normal polarity. 1 = The polarity of the Attention and Power indicators presented to the IOX is inverted.	0b	RW
01:00	Reserved	**Reserved**	0b	

13.4.1.27 Offset 45h: VSCMD1 – Vendor Specific Command Byte 1 Register

This register is for vendor specific commands.

Table 326. Offset 45h: VSCMD1 – Vendor Specific Command Byte 1 Register

Device: 2	***Function:*** 0
Offset: *45h*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	0b	
03	CTOD	**Completion TO Timer Disable:** 0 = The completion Timeout Timer is enable. 1 = The completion Timeout Timer is disabled.	0b	RW
02	HGD	**Hot plug GPE Disable:** 0 = Enables reporting of hot plug interrupts via the legacy GPE mechanism. 1 = Disables reporting of hot plug interrupts via the legacy GPE mechanism. This bit must be set when hot plug interrupts are to be reported via the interrupt (INTx or MSI) signaling mechanism.	0b	RW
01	TCLE	**Training Control Loopback Enable:** 0 = Disabled 1 = Enabled - If this bit is a 1 when the TS1/TS2 ordered-sets are transmitted, the “Enable Loopback” bit is set in the training control symbol	0b	RW
00	PMETOR	**PME Turn Off Request:** 0 = Cleared by hardware when the acknowledge is returned from the link. The bit is also cleared when the link layer is in the DL_down state. 1 = Set by software	0b	RWS

13.4.1.28 Offset 46h: VSSTS0 – Vendor Specific Status Byte 0 Register

This register is for vendor specific status.

Table 327. Offset 46h: VSSTS0 – Vendor Specific Status Byte 0 Register (Sheet 1 of 2)

Device: 2	***Function:*** 0
Offset: *46h*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	00h	
03	SMB Busy	**SMB Busy:** This bit indicates that the bus is busy, but that this master is not involved in the traffic. 0 = Ready 1 = Busy	0b	RO

Table 327. Offset 46h: VSSTS0 – Vendor Specific Status Byte 0 Register (Sheet 2 of 2)

Device: 2	**Function:** 0
Offset: 46h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
02	Reserved	Reserved	0b	
01	HPCB	**Hot Plug Controller Busy:** This bit is true when the common controller is busy servicing requests, basically non-idle of the master FSM. 0 = Ready 1 = Busy	0b	RO
00	HPIC	Hot Plug Initialize Complete This bit is read to determine if the hot plug controller has finished initializing itself, after it has been enabled through the hot plug capable bit in the PCI Express slot capabilities register. 0 = Hot Plug not initialized 1 = Hot Plug initialized	0b	RO

13.4.1.29 Offset 47h: VSSTS1 – Vendor Specific Status Byte 1 Register

This register is for vendor specific status.

Table 328. Offset 47h: VSSTS1 – Vendor Specific Status Byte 1 Register

Device: 2	**Function:** 0
Offset: 47h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07:02	Reserved	Reserved	00h	
01	LA	**Link Active:** Bit reports whether transactions are being sent or aborted by the downstream transaction control, which is determined by the “link_active” status from the link layer reflected in this status bit. 0 = Link Not Active 1 = Link Active	0b	RO
00	PMETOA	**PME Turn Off Acknowledge:** 0 = Software writes a 1 to this bit to clear it. The bit will also be cleared when the link layer is in the DL_down state. 1 = Set by hardware when PMETOR (Table 326 on page 430) is ON and the acknowledge is returned from the link. When this bit (PMETOA) is set, PMETOR (Table 326 on page 430) clears.	0b	RWC

13.4.1.30 Offset 50h: PMCAPID – Power Management Capabilities Structure Register

This register identifies the capability structure and points to the next structure.

Table 329. Offset 50h: PMCAPID – Power Management Capabilities Structure Register

Device: *2*	***Function:*** *0*
Offset: *50h*	***Size:*** *8 bit*
Default Value: *01h*	

Bits	Name	Description	Reset Value	Access
07:00	CAP_ID	This field has the value 01h to identify the CAP_ID assigned by the PCI SIG for vendor dependent capability pointers.	01h	RO

13.4.1.31 Offset 51h: PMNPTR – Power Management Next Capabilities Pointer Register

This register identifies the capability structure and points to the next structure.

Table 330. Offset 51h: PMNPTR – Power Management Next Capabilities Pointer Register

Device: *2*	***Function:*** *0*
Offset: *51h*	***Size:*** *8 bit*
Default Value: *58h*	

Bits	Name	Description	Reset Value	Access
07:00	NCR	**Next Capability Pointer:** This field points to the next Capability ID in this device which is the MSI.	58h	RO

13.4.1.32 Offset 52 - 53h: PMCAPA – Power Management Capabilities Register

This register identifies the capabilities for PM.

Table 331. Offset 52 - 53h: PMCAPA – Power Management Capabilities Register

Device: 2 ***Function:*** 0
Offset: *52 - 53h* ***Size:*** *16 bit*
Default Value: *C822h*

Bits	Name	Description	Reset Value	Access
15:11	PMES_PMCAPA	**PME Support:** Identifies power states which assert PME. Bits 15, 14 and 11 must be set to '1' for PCI-PCI bridge structures representing ports on root complexes. The definition of these bits is taken from the *PCI Bus Power Management Interface Specification Revision 1.1.* bit(11) XXXX1b PME# can be asserted from D0 bit(12) XXX1Xb PME# can be asserted from D1 (IMCH does not support) bit(13) XX1XXb PME# can be asserted from D2 (IMCH does not support) bit(14) X1XXXb PME# can be asserted from D3 hot bit(15) 1XXXXb PME# can be asserted from D3 cold	11001b	RO
10	D2S	**D2 Support: This bit is** hardwired to '0' to indicate the power management state D2 is not supported.	0b	RO
09	D1S	**D1 Support: This bit is** hardwired to '0' to indicate the power management state D1 is not supported.	0b	RO
08:06	AUXCC	**AUXC current:** AUX current required for function. Hardwired to 000b to indicate a self-powered device.	000b	RO
05	DSI	**DSI:** Device Specific Initialization is required.	1b	RO
04	Reserved	Reserved	0b	
03	PMEC	**PME Clock: This bit is** hardwired to '0' to indicate no PCI Clock is required for PME.	0b	RO
02:00	VER	**Version:** Hardwired to 010b to indicate compliance with *PCI Bus Power Management Interface Specification, Rev 1.1.*	010b	RO

13.4.1.33 Offset 54 - 55h: PMCSR – Power Management Status and Control Register

This register contains the control and status bits for power management.

Table 332. Offset 54 - 55h: PMCSR – Power Management Status and Control Register (Sheet 1 of 2)

Device: 2 ***Function:*** 0
Offset: *54 - 55h* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15	PMES_PMCSR	**PME Status [STICKY]: This bit is** hardwired to '0' to Indicate this field is not supported by IMCH. This bit is sticky.	0b	RO
14:09	Reserved	Reserved	00b	

Table 332. Offset 54 - 55h: PMCSR – Power Management Status and Control Register (Sheet 2 of 2)

Device: 2	**Function:** 0
Offset: 54 - 55h	**Size:** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
08	PMEE	**PME Enable [STICKY]:** Controls PME# assertion. This bit is sticky through reset. Writes to this field have no effect. This bit is sticky. 0 = This device does not assert PME# 1 = Enables this device to assert PME#	0b	RW
07:02	Reserved	Reserved	00h	
01:00	PS	**Power State:** Since the PCI Express bridge device supports only the D0 state, writes to this field have no effect.	00b	RO

13.4.1.34 Offset 56h: PMCSRBSE – Power Management Status and Control Bridge Extensions Register

This register identifies the status and control bridge extensions for power management.

Table 333. Offset 56h: PMCSRBSE – Power Management Status and Control Bridge Extensions Register

Device: 2	**Function:** 0
Offset: 56h	**Size:** 8 bit
Default Value: 00h	

Bits	Name	Description	Reset Value	Access
07	BP_CCE	**Bus Power/Clock Control Enable:** Hardwired to 0.	0b	RO
06	B2B3S	**B2/B3 Support:** Not applicable.Hardwired to 0.	0b	RO
05:00	Reserved	Reserved	00h	

13.4.1.35 Offset 58h: MSICAPID – MSI Capabilities Structure Register

This register identifies the MSI capability structure.

Table 334. Offset 58h: MSICAPID – MSI Capabilities Structure Register

Device: 2	**Function:** 0
Offset: 58h	**Size:** 8 bit
Default Value: 05h	

Bits	Name	Description	Reset Value	Access
07:00	CAP_ID	This field has the value 05h to identify the CAP_ID assigned by the PCI SIG for a message signaled interrupts capability list.	05h	RO

13.4.1.36 Offset 59h: MSINPTR – MSI Next Capabilities Pointer Register

This register points to the next capability structure.

Table 335. Offset 59h: MSINPTR – MSI Next Capabilities Pointer Register

Device: 2	***Function:*** 0
Offset: *59h*	***Size:*** *8 bit*
Default Value: *64h*	

Bits	Name	Description	Reset Value	Access
07:00	MSI_NCP	**Next Capability Pointer:** This field points to the next Capability ID in this device, which is the Hot Plug Controller.	64h	RO

13.4.1.37 Offset 5A - 5Bh: MSICAPA – MSI Capabilities Register

The PCI Express controller generates upstream interrupt messages using MSI to the processor bypassing IOxAPIC. The MSI is generated by a Memory Write to address 0FEEx_xxxxh. Three 32-bit registers are required in the PCI Express controller to support this mechanism. The default values of these registers are compatible to the default value of IOxAPIC. The software can reprogram these registers to required values. The three registers are MSI Control Register (MSICR), MSI Address Register (MSIAR) and MSI Data Register (MSIDR). Depending on system requirements each PCI Express channel can have a MSI block (provides better flexibility) or the PCI Express controller as a whole can have one MSI block and all channels raise hardware interrupts to this block.

The MSI Control Register (MSICR) contains all the information related to the capability of PCI Express MSI interrupts. The MSICR register has been broken down into its components, MSICAPID, MSINPTR, and MSICAPA for purposes of separate register definitions.

Table 336. Offset 5A - 5Bh: MSICAPA – MSI Capabilities Register

Device: 2	***Function:*** 0
Offset: *5A - 5Bh*	***Size:*** *16 bit*
Default Value: *0002h*	

Bits	Name	Description	Reset Value	Access
15:08	Reserved	Reserved	00h	
07	64AC	**Indicates 64-bit Address Capable:** Hardwired to '0' to indicate that the PCI Express bridge is capable of 32-bit MSI addressing.	0b	RO
06:04	MME	**Multiple Message Enable:** The software writes this field to indicate the number of allocated messages, which is aligned to a power of two. When MSI is enabled, the software allocates at least one message to the device.	0h	RW
03:01	MMC	**Multiple Message Capable:** PCI Express requests a capability for two messages by initializing this field to a value of 001b.	001b	RO
00	MSIE	**MSI Enable:** Software sets this bit to select the method of interrupt delivery. If no interrupts are enabled, software must poll for status since no interrupts of either type are generated. 0 = Legacy interrupts are generated. 1 = Message Signaled Interrupts (MSI) are generated.	0b	RW

13.4.1.38 Offset 5C - 5Fh: MSIAR – MSI Address for PCI Express Register

The MSI Address Register (MSIAR) contains all the address related information to route MSI interrupts.

Table 337. Offset 5C - 5Fh: MSIAR – MSI Address for PCI Express Register

Device: 2 ***Function:*** 0
Offset: *5C - 5Fh* ***Size:*** *32 bit*
Default Value: *FEE0_0000h*

Bits	Name	Description	Reset Value	Access
31:20	MSIA	**Address:** Most significant 12 bits of 32-bit address.	FEEh	RW
19:12	DESID	**Destination ID:** Should reflect the 63:56 bits of IOxAPIC redirection table entry. The IMCH may substitute other values in this field when redirecting to the System Bus.	00h	RW
11:04	EXDID	**Extended Destination ID:** Should reflect the 55:48 bits of IOxAPIC redirection table entry.	00h	RW
03	RH	**Redirection Hint:** Used by the IMCH to allow the interrupt message to be redirected. 0 = Direct 1 = Redirect	0b	RW
02	DMMSIA	**Destination Mode:** Used only if Redirection Hint is set to '1'. 0 = Physical 1 = Logical	0b	RW
01:00	Reserved	Reserved	0b	

13.4.1.39 Offset 60 - 61h: MSIDR – MSI Data Register

The MSI Data Register (MSIDR) contains all the data related information to route MSI interrupts.

Table 338. Offset 60 - 61h: MSIDR – MSI Data Register (Sheet 1 of 2)

Device: 2 ***Function:*** 0
Offset: *60 - 61h* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15	TM	**Trigger Mode:** Definition is the same as the corresponding bit in the I/O Redirection Table for that interrupt. 0 = Edge 1 = Level	0b	RW
14	DVS	**Delivery Status:** If using edge-triggered interrupts, this is always a 1, since only assertion is sent. If using level-triggered interrupts, then this bit indicates the state of the interrupt input.	0b	RW
13:12	Reserved	Reserved	0b	

Table 338. Offset 60 - 61h: MSIDR – MSI Data Register (Sheet 2 of 2)

Device: 2		**Function:** 0		
Offset: 60 - 61h		**Size:** 16 bit		
Default Value: 0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
11	DMMSID	**Destination Mode:** Same as bit 2 of MSIAR. 0 = Physical 1 = Logical	0b	RW
10:08	DELM	**Delivery Mode:** Same as the corresponding bits in the I/O Redirection Table for that interrupt. 000 = Fixed 100 = NMI 001 = Lowest Priority 101 = INIT 010 = SMI/PMI 110 = Reserved 011 = Reserved 111 = ExtINT	0h	RW
07:00	IV	**Interrupt Vector:** Same as the corresponding bits in the I/O Redirection Table for that interrupt.	00h	RW

13.4.1.40 Offset 64h: PEACAPID – PCI Express Features Capabilities ID Register

This register identifies the PCI Express features capability structure.

Table 339. Offset 64h: PEACAPID – PCI Express Features Capabilities ID Register

Device: 2		**Function:** 0		
Offset: 64h		**Size:** 8 bit		
Default Value: 10h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:00	CAP_ID	This field has the value 10h to identify the CAP_ID assigned by the PCI SIG for PCI Express capability structure.	10h	RO

13.4.1.41 Offset 65h: PEANPTR – PCI Express Next Capabilities Pointer Register

This register identifies the next PCI Express capability structure.

Table 340. Offset 65h: PEANPTR – PCI Express Next Capabilities Pointer Register

Device: 2		**Function:** 0		
Offset: 65h		**Size:** 8 bit		
Default Value: 00h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:00	PEA_NCP	**Next Capability Pointer:** This field contains that value 00b to indicate that there are no additional capability structures.	00h	RO

13.4.1.42 Offset 66 - 67h: PEACAPA – PCI Express Features Capabilities Register

This register identifies PCI Express device type and associated capabilities.

Table 341. Offset 66 - 67h: PEACAPA – PCI Express Features Capabilities Register

Device: 2	***Function:*** 0
Offset: *66 - 67h*	***Size:*** *16 bit*
Default Value: *0041h*	

Bits	Name	Description	Reset Value	Access
15:14	Reserved	Reserved	00b	
13:09	CIMN	**Capability Interrupt Message Number:** If the function is allocated more than one MSI interrupt number, this field contains the offset between the base Message Data and the MSI Message that is generated when any of the status bits in either the Slot Status or Root Port Status registers of this capability structure are set. Hardware updates this field so that it is correct with the number of MSI Messages assigned to the device (based on the setting of the Multiple Message Enable bits in the MSI Capabilities register).	00000b	RO
08	SIMP	**Slot Implemented:** BIOS must set this bit at boot time if the PCI Express link associated with this port is connected to a slot (as compared to being connected to a motherboard component, or being disabled). 0 = Slot not implemented. 1 = Slot implemented.	0b	RWO
07:04	DPT	**Device/Port Type:** Hardwired to a value of "4" hex to indicate a root port.	4h	RO
03:00	CAPV	**Capability Version:** Hardwired to 1h to indicate compliance with the *PCI Express* Interface Specification, Rev 1.0a*.	1h	RO

13.4.1.43 Offset 68 - 6Bh: PEADEVCAP – PCI Express Device Capabilities Register

This register identifies the device capabilities for PCI Express.

Table 342. Offset 68 - 6Bh: PEADEVCAP – PCI Express Device Capabilities Register (Sheet 1 of 2)

Device: 2	***Function:*** 0
Offset: *68 - 6Bh*	***Size:*** *32 bit*
Default Value: *0000_0001h*	

Bits	Name	Description	Reset Value	Access
31:28	Reserved	Reserved	0h	
27:26	CSPLS	**Captured Slot Power Limit Scale (Upstream Ports Only):** Specifies the scale used for the Slot Power Limit Value. 00b = 1.0x (25.5 – 255) 01b = 0.1x (2.55 – 25.5) 10b = 0.01x (0.255 – 2.55) 11b = 0.001x (0.0 – 0.255)	00b	RO
25:18	CSPLV	**Captured Slot Power Limit Value (Upstream Ports Only):** In combination with the Slot Power Limit Scale value, this register specifies the upper limit on power supplied by slot. Power limit (in watts) calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field. This value is set by the Set_Slot_Power_Limit.	00h	RO

Table 342. Offset 68 - 6Bh: PEADEVCAP – PCI Express Device Capabilities Register (Sheet 2 of 2)

Device: 2	**Function:** 0
Offset: 68 - 6Bh	**Size:** 32 bit
Default Value: 0000_0001h	

Bits	Name	Description	Reset Value	Access
17:06	Reserved	Reserved	000b	
05	ETFS	**Extended Tag Field Supported:** Hardwired to 0b, indicating 5 bits, as required for a Root port.	0b	RO
04:03	PFS	**Phantom Functions Supported:** Hardwired to 00b as required for Root ports, indicating that devices may implement all function numbers.	00b	RO
02:00	MPSS	**Max Payload Size Supported:** Hardwired to 001b to indicate a maximum 256B payload size. Note that this refers to an inbound payload size, since the outbound payload size is restricted to a cacheline size to a value of 64 B.	001b	RO

13.4.1.44 Offset 6C - 6Dh: PEADEVCTL – PCI Express Device Control Register

This register identifies PCI Express device specific parameters.

Table 343. Offset 6C - 6Dh: PEADEVCTL – PCI Express Device Control Register (Sheet 1 of 2)

Device: 2	**Function:** 0
Offset: 6C - 6Dh	**Size:** 16 bit
Default Value: 0000h	

Bits	Name	Description	Reset Value	Access
15	Reserved	Reserved	0b	
14:12	MRRS	**Max Read Request Size:** This field sets maximum read request size for the device as a requester. The IMCH does not generate read requests with size exceeding the set value. Defined encodings for this field are: 000b = 128 B 100b = 2 Kbyte 001b = 256 B 101b = 4 Kbyte 010b = 512 B 110b = Reserved 011b = 1 Kbyte 111b = Reserved	000b	RW
11	ENS	**Enable No Snoop:** Permits the device to set the No Snoop bit in the Requester Attributes of transactions that do not require hardware enforced cache coherency. Even when this bit is set, the device can only set the No Snoop attribute on a transaction when the address of the transaction is not stored on any cache in the system. 0 = Disable 1 = Enable Software override on usage of the "No Snoop" attribute. The IMCH hard-wires this bit to 0, as it never issues transactions with that attribute set.	0b	RO
10	AUXPPE	**Auxiliary (AUX) Power PM Enable:** Not Applicable.	0b	RO
09	PFE	**Phantom Functions Enable:** Not Applicable.	0b	RO
08	ETFE	**Extended Tag Field Enable:** Not Applicable to a Root port.	0b	RO

Table 343. Offset 6C - 6Dh: PEADEVCTL – PCI Express Device Control Register (Sheet 2 of 2)

Device: 2 **Function:** 0
Offset: 6C - 6Dh **Size:** 16 bit
Default Value: 0000h

Bits	Name	Description	Reset Value	Access
07:05	MAXPS	**Max Payload Size:** This field sets maximum TLP payload size for the device. As a receiver, the device must handle TLPs as large as the set value; as transmitter, the device must not generate TLPs exceeding the set value. Permissible values that can be programmed are indicated by the max_payload_size supported in the device capabilities register. NOTE: Encodings above 256B are not supported. RW functionality is only maintained for compliance testing of all register bits. Defined encodings for this field are: 000b = 128 B 100b = 2 KByte 001b = 256 B 101b = 4 KByte 010b = 512 B 110b = Reserved 011b = 1 KByte 111b = Reserved	000b	RW
04	ERO	**Enable Relaxed Ordering:** Hard-wired to "0" in the IMCH, as no such transaction attributes are ever used on outbound requests.	0b	RO
03	URRE	**Unsupported Request Reporting Enable:** This bit enables reporting of unsupported requests when set. Default is "0" with reporting disabled. Note that the reporting of error messages (ERR_CORR, ERR_NONFATAL, ERR_FATAL) received by Root Port is controlled exclusively by Root Port Control Register. 0 = Disable reporting of Unsupported Request errors 1 = Enable reporting of Unsupported Request errors	0b	RW
02	FERE	**Fatal Error Reporting Enable:** This bit controls the reporting of fatal errors. Note that the reporting of fatal errors is internal to the root. No external ERR_FATAL message is generated. PCICMD[SERRE] when set can also enable reporting of both internal and external errors to be reported. 0 = Disable fatal error reporting 1 = Enable fatal error reporting	0b	RW
01	NFERE	**Non-fatal Error Reporting Enable:** This bit controls the reporting of nonfatal errors. Note that the reporting of nonfatal errors is internal to the root. No external ERR_NONFATAL message is generated. PCICMD[SERRE] when set can also enable reporting of both internal and external errors to be reported. 0 = Disable nonfatal error reporting 1 = Enable nonfatal error reporting	0b	RW
00	CERE	**Correctable Error Reporting Enable:** This bit controls the reporting of correctable errors. Note that the reporting of correctable errors is internal to the root. No external ERR_CORR message is generated. 0 = Disable correctable error reporting 1 = Enable correctable error reporting	0b	RW

13.4.1.45 Offset 6E - 6Fh: PEADEVSTS – PCI Express Device Status Register

This register provides information about PCI Express device specific parameters.

Table 344. Offset 6E - 6Fh: PEADEVSTS – PCI Express Device Status Register

Device: 2 ***Function:*** 0
Offset: *6E - 6Fh* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:06	Reserved	Reserved	000h	
05	TP	**Transactions Pending:** Indicates that the device has transactions pending. 0 = Cleared by hardware only when all pending transactions (including completions for any outstanding non-posted requests on any used virtual channel) have been completed. 1 = Set by hardware to indicate that transactions are pending (including completions for any outstanding non-posted requests for all used Traffic Classes).	0b	RO
04	APD	**AUX Power Detected:** Not Applicable.	0b	RO
03	URS	**Unsupported Request Detected:** Indicates that an Unsupported Request has been detected. This bit is set upon Unsupported Request detection regardless of whether or not error reporting is enabled in the Device Control register. Software clears this bit by writing a '1' to the bit location. 0 = No Unsupported Request detected 1 = Unsupported Request detected	0b	RWC
02	FED	**Fatal Error Detected:** Indicates that a fatal error has been detected. This bit is set upon fatal error detection regardless of whether or not error reporting is enabled in the Device Control register. Software clears this bit by writing a '1' to the bit location. 0 = No fatal error detected 1 = Fatal error detected	0b	RWC
01	NFED	**Non-fatal Error Detected:** Indicates that a nonfatal error has been detected. This bit is set upon nonfatal error detection regardless of whether or not error reporting is enabled in the Device Control register. Software clears this bit by writing a '1' to the bit location. 0 = No nonfatal error detected 1 = Nonfatal error detected	0b	RWC
00	CED	**Correctable Error Detected:** Indicates that a correctable error has been detected. This bit is set upon correctable error detection regardless of whether or not error reporting is enabled in the Device Control register. Software clears this bit by writing a '1' to the bit location. 0 = No correctable error detected 1 = Correctable error detected	0b	RWC

13.4.1.46 Offset 70 - 73h: PEALNKCAP – PCI Express Link Capabilities Register

This register identifies PCI Express link specific capabilities.

Table 345. Offset 70 - 73h: PEALNKCAP – PCI Express Link Capabilities Register

Device: *2*	***Function:*** *0*
Offset: *70 - 73h*	***Size:*** *32 bit*
Default Value: *0203_E481h*	

Bits	Name	Description	Reset Value	Access
31:24	PN	**Port Number:** This field indicates the PCI Express port number for the associated PCI Express link.	02h	RO
23:18	Reserved	Reserved	00h	
17:15	L1EL	Intel® 3100 Chipset **does not support L1.**Not Applicable	111b	RO
14:12	L0EL	Intel® 3100 Chipset does not support L0s. Not Applicable	110b	RO
11:10	ASPM	Intel® 3100 Chipset does not support L0s or L1. **Active State PM:** 01 L0s Entry Supported (Per PCI Express spec, L0s must be supported, but Intel® 3100 Chipset does not support L0s.)	01b	RO
09:04	MLW	**Maximum Link Width:** This field indicates the maximum width of the PCI Express link. The Read function is only allowed after Software/BIOS initialized the bits. Device 2 reports a value of 001000b, indicating a maximum link width of x8. However, if two separate devices are connected to port A (Device 2) and port A1 (Device 3), the maximum link width for both ports is x4.	00_1000b	RWO
03:00	MLS	**Maximum Link Speed:** Hardwired to a value of 1h, to indicate a maximum link speed of 2.5 Gbits/s.	0001b	RO

13.4.1.47 Offset 74 - 75h: PEALNKCTL – PCI Express Link Control Register

This register controls PCI Express link specific parameters.

Table 346. Offset 74 - 75h: PEALNKCTL – PCI Express Link Control Register

Device: 2 ***Function:*** 0
Offset: *Offset 74 - 75h* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:08	Reserved	Reserved	00h	
07	ES	Intel® 3100 Chipset does not support L0s or L1. This bit only has an impact on devices transitioning to/from L0. **Extended Synch:** Provides external devices monitoring the link with additional time for to achieve bit and symbol lock before the link enters L0 state and resumes communication. 0 = Normal 1 = Reserved.	0b	RW
06	CCC	**Common Clock Configuration:** 0 = This component and the component at the opposite end of the link are operating with asynchronous reference clocks. 1 = This component and the component at the opposite end of the link are operating with a distributed common reference clock.	0b	RW
05	RL	**Retrain Link:** 0 = Link retraining not initiated. This bit always returns 0 when read. 1 = Link retraining initiated ***Note:*** Link retraining does not force a "Link Down" condition, it merely invokes "recovery."	0b	WO
04	LD	**Link Disable:** Disables/Enables the associated PCI Express link. 0 = Enable 1 = Disable	0b	RW
03	RCB	**Read Request Return parameter "R" Control:** Hardwired to '0', indicating "RCB" capability of 64B. This is also known as Read Completion Boundary.	0b	RO
02	Reserved	Reserved	0b	
01:00	ASLPMC	L0s is not supported, and ASPM should never be turned on. **Leave disabled (00b default).** External PCI Express devices should never request entrance into L0s. If it does, undefined behavior will result. External device must set ASPM control register to "disable", or 00b. **Active State Link PM Control:** Controls the level of active state power management supported on the associated PCI Express link. Defined encodings are: 00b Disabled 01b L0s Entry Supported 10b Reserved 11b Reserved	00b	RW

13.4.1.48 Offset 76 - 77h: PEALNKSTS – PCI Express Link Status Register

This register provides information about PCI Express Link specific parameters.

Table 347. Offset 76 - 77h: PEALNKSTS – PCI Express Link Status Register

Device: *2*	***Function:*** *0*
Offset: *76 - 77h*	***Size:*** *16 bit*
Default Value: *1001h*	

Bits	Name	Description	Reset Value	Access
15:13	Reserved	Reserved	000b	
12	SCC	**Slot Clock Configuration:** This bit indicates that the component uses the same physical reference clock that the platform provides on the connector.The Read function is only allowed after Software/BIOS initialized the bit. 0 = The component in the slot uses an independent reference clock, irrespective of the presence of a reference on the connector. 1 = The component in the slot uses the same physical reference clock provided on the connector.	1b	RWO
11	LT	**Link Training:** This read-only bit indicates that Link training is in progress (Physical Layer LTSSM in Configuration or Recovery state); hardware clears this bit once Link Training is successfully trained to the L0 state. 0 = Cleared by hardware once link training is complete 1 = Set by hardware when link training is in progress	0b	RO
10	LTE	**Link Training Error:** This bit is set on transition from configuration or recovery to detect and is cleared when link layer reports DL_UP. 0 = No Link Training Error 1 = Link Training Error occurred	0b	RO
09:04	NW	**Negotiated Width:** Note that reset value is reserved, and this field remains undefined until bit 11 (Link Training) has been cleared by hardware. If training never completes, this field remains undefined. 000001b = x1 000100b = x4 001000b = x8 All other encodings are reserved.	00h	RO
03:00	LS	**Link Speed:** Value of 1h indicates 2.5 GBytes/s link.	1h	RO

13.4.1.49 Offset 78 - 7Bh: PEASLTCAP – PCI Express Slot Capabilities Register

This register identifies PCI Express slot specific capabilities.

Table 348. Offset 78 - 7Bh: PEASLTCAP – PCI Express Slot Capabilities Register (Sheet 1 of 2)

Device: 2	***Function:*** 0
Offset: *78 - 7Bh*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
31:19	PSN	**Physical Slot Number:** This hardware initialized field indicates the physical slot number attached to this port. This field must be hardware initialized to a value that assigns a slot number that is globally unique within the chassis. This field must be initialized to 0 for ports connected to devices that are integrated on the motherboard.	000h	RWO
18:17	Reserved	Reserved	00b	
16:15	SPLS	**Slot Power Limit Scale** Specifies the scale used for the Slot Power Limit Value. 00b = 1.0X (25.5–255) 01b = 0.1X (2.55–25.5) 10b = 0.01X (0.255–2.55) 11b = 0.001X (0.0–0.255)	00b	RWO
14:07	SPLV	**Slot Power Limit Value** In combination with the Slot Power Limit Scale value, this register specifies the upper limit on power supplied by slot. Power limit (in watts) calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field. This field must be programmed at boot. Writing to this field triggers a Set_Slot_Power_Limit inband PCI Express* message.	00h	RWO
06	HOTPC	**Hot plug capable:** This bit is used as hot plug enable for a controller. This register is not valid for Device 3 because hot plug is only valid on Device 2. SW must not attempt to write to this register in Device 3. 0 = Indicates that this slot is not capable of supporting hot plug operations. 1 = Indicates that this slot is capable of supporting hot plug operations.	0b	RWO
05	HOTPS	**Hot plug Surprise:** Not supported. This bit is hardwired to 0.	0b	RO
04	PIP	**Power Indicator Present:**. This register is not valid for Device 3 because hot plug is only valid on Device 2. SW must not attempt to write to this register in Device 3. 0 = Not present. 1 = Indicates that a Power Indicator is implemented on the chassis for this slot.	0b	RWO
03	AIP	**Attention Indicator Present:** This register is not valid for Device 3 because hot plug is only valid on Device 2. SW must not attempt to write to this register in Device 3. 0 = Not present. 1 = Indicates that an Attention Indicator is implemented on the chassis for this slot.	0b	RWO
02	MSP	**MRL Sensor Present:** This register is not valid for Device 3 because hot plug is only valid on Device 2. SW must not attempt to write to this register in Device 3. 0 = Not present. 1 = Indicates that an MRL Sensor is implemented on the chassis for this slot.	0b	RWO

Table 348. Offset 78 - 7Bh: PEASLTCAP – PCI Express Slot Capabilities Register (Sheet 2 of 2)

Device: 2	**Function:** 0
Offset: 78 - 7Bh	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
01	PCP	**Power Controller Present:** This register is not valid for Device 3 because hot plug is only valid on Device 2. SW must not attempt to write to this register in Device 3. 0 = Not present. 1 = Indicates that a Power Controller is implemented for this slot.	0b	RWO
00	ABP	**Attention Button Present:** This register is not valid for Device 3 because hot plug is only valid on Device 2. SW must not attempt to write to this register in Device 3. 0 = Not present. 1 = Indicates than an Attention Button is implemented on the chassis for this slot.	0b	RWO

13.4.1.50 Offset 7C - 7Dh: PEASLTCTL – PCI Express Slot Control Register

This register controls PCI Express Slot specific parameters.

This register is not valid for Device 3 because hot plug is only valid on Device 2. SW must not attempt to write to this register in Device 3.

Table 349. Offset 7C - 7Dh: PEASLTCTL – PCI Express Slot Control Register (Sheet 1 of 2)

Device: 2	**Function:** 0
Offset: 7C - 7Dh	**Size:** 16 bit
Default Value: 01C0h	

Bits	Name	Description	Reset Value	Access
15:11	Reserved	Reserved	00h	
10	PCC	**Power Controller Control:** When read this register returns the current state of the power applied to the slot; when written sets the power state of the slot per the defined encodings. 0 = Power On 1 = Power Off The hardware only supports hot plug functionality on Device 2 so this read/write bit affects no logic on other devices.	0b	RW
09:08	PIC	**Power Indicator Control:** Read to this register return the current state of the Power Indicator; writes to this register set the Power Indicator. Defined Encodings are: 00b Reserved 01b On 10b Blink 11b Off Writes to this register also cause the Port to send the appropriate POWER_INDICATOR_* messages. The hardware only supports hot plug functionality on Device 2 so this read/write bit affects no logic on other devices. ***Note:*** Reserved values must not be used. Undefined behavior will result if they are used.	01b	RW

Table 349. Offset 7C - 7Dh: PEASLTCTL – PCI Express Slot Control Register (Sheet 2 of 2)

Device: 2	**Function:** 0
Offset: 7C - 7Dh	**Size:** 16 bit
Default Value: 01C0h	

Bits	Name	Description	Reset Value	Access
07:06	AIC	**Attention Indicator Control:** Reads to this register returns the current state of the Attention Indicator. Writes to this register set the Attention Indicator. Defined Encodings are: 00b Reserved 01b On 10b Blink 11b Off Writes to this register also cause the Port to send the appropriate ATTENTION_INDICATOR_* messages. The hardware only supports hot plug functionality on Device 2 so this read/write bit affects no logic on other devices. ***Note:*** Reserved values must not be used. Undefined behavior will result if they are used.	11b	RW
05	HPIE	**Hot plug Interrupt Enable:** The hardware only supports hot plug functionality on Device 2 so this read/write bit affects no logic on other devices. 0 = Disable 1 = Enables generation of hot plug interrupts on enabled hot plug events.	0b	RW
04	CCIE	**Command Complete Interrupt Enable:** The hardware only supports hot plug functionality on Device 2 so this read/write bit affects no logic on other devices. 0 = Disable 1 = Enables the generation of hot plug interrupt when a command is completed by the hot plug controller. The command affected by this bit is any write to the slot control register.	0b	RW
03	PDCIE	**Presence Detect Changed Interrupt Enable:** The hardware only supports hot plug functionality on Device 2 so this read/write bit affects no logic on other devices. 0 = Disable 1 = Enables the generation of hot plug interrupt or wake message on an presence detect changed event.	0b	RW
02	MSCIE	**MRL Sensor Changed Interrupt Enable:** The hardware only supports hot plug functionality on Device 2 so this read/write bit affects no logic on other devices. 0 = Disable 0 = Enables the generation of hot plug interrupt on an MRL (Mechanical Retention Latch) sensor changed event.	0b	RW
01	PFDIE	**Power Fault Detected Interrupt Enable:** The hardware only supports hot plug functionality on Device 2 so this read/write bit affects no logic on other devices. 0 = Disable 1 = This bit when set enables the generation of hot plug interrupt on a power fault event.	0b	RW
00	ABDIE	**Attention Button Pressed Interrupt Enable:** The hardware only supports hot plug functionality on Device 2 so this read/write bit affects no logic on other devices. 0 = Disable 1 = Enables the generation of hot plug interrupt on an attention button pressed event.	0b	RW

13.4.1.51 Offset 7E - 7Fh: PEASLTSTS – PCI Express Slot Status Register

This register provides information about PCI Express Slot specific parameters.

Table 350. Offset 7E - 7Fh: PEASLTSTS – PCI Express Slot Status Register

Device: 2	***Function:*** 0
Offset: *7E - 7Fh*	***Size:*** *16 bit*
Default Value: *0040h*	

Bits	Name	Description	Reset Value	Access
15:07	Reserved	Reserved	000h	
06	PDS	**Presence Detect State:** This bit indicates the presence of a card in the slot. The bit reflects the Presence Detect status determined via in-band mechanism or via Present Detect pins on the slot itself. This register is required if a slot is implemented. 0 = Slot empty 1 = Card present in slot ***Note:***	1b	RO
05	MSS	**MRL Sensor State:** This register reports the status of the MRL sensor if it is implemented. 0 = MRL Closed 1 = MRL Open This register is not valid for Device 3 because hot plug is only valid on Device 2. SW must not attempt to write to this register in Device 3.	0b	RO
04	COMC	**Command Completed:** This register is not valid for Device 3 because hot plug is only valid on Device 2. SW must not attempt to write to this register in Device 3. 0 = Not completed 1 = The hot plug controller completed an issued command and is ready to accept the next command. The command completed bit will be set for any write to the slot control register.	0b	RWC
03	PDC	**Presence Detect Changed:** This register is not valid for Device 3 because hot plug is only valid on Device 2. SW must not attempt to write to this register in Device 3. 0 = No change detected. 1 = A Presence Detect change is detected.	0b	RWC
02	MSC	**MRL Sensor Changed:** This register is not valid for Device 3 because hot plug is only valid on Device 2. SW must not attempt to write to this register in Device 3. 0 = No change detected. 1 = A MRL Sensor state change is detected.	0b	RWC
01	PFD	**Power Fault Detected:** This register is not valid for Device 3 because hot plug is only valid on Device 2. SW must not attempt to write to this register in Device 3. 0 = No fault detected. 1 = The Power Controller detects a power fault at this slot.	0b	RWC
00	ATBP	**Attention Button Pressed:** This register is not valid for Device 3 because hot plug is only valid on Device 2. SW must not attempt to write to this register in Device 3. 0 = Attention button is not pressed. 1 = The attention button is pressed.	0b	RWC

13.4.1.52 Offset 80 - 83h: PEARPCTL – PCI Express Root Port Control Register

This register enables the forwarding of error messages based on messages received.

Table 351. Offset 80 - 83h: PEARPCTL – PCI Express Root Port Control Register

Device: 2	***Function:*** 0
Offset: *80 - 83h*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
31:04	Reserved	Reserved	000_0000h	
03	EPI	**Enable PME Interrupt:** Enables/disables interrupt generation upon receipt of a PME message as reflected in the PME Status register bit. A PME interrupt is also generated if the PME Status register bit is already set when this bit is set from a cleared state. 0 = Disable PME interrupt (PME#) generation 1 = Enable PME interrupt (PME#) generation	0b	RW
02	ESEFE	**Enable system error on Fatal Error:** Controls the Root Complex's response to fatal errors reported by any of the devices in the hierarchy associated with this Root Port. System error generation based on fatal errors also enabled by PCICMD[SERR]. 0 = Disable System Error generation in response to fatal errors reported on this port. 1 = Enable System Error generation in response to fatal errors reported on this port.	0b	RW
01	ESENFE	**Enable system error on Non-Fatal Error:** Controls the Root Complex's response to nonfatal errors reported by any of the devices in the hierarchy associated with this Root Port. System error generation based on non-fatal errors also enabled by PCICMD[SERR]. 0 = Disable System Error generation in response to nonfatal errors reported on this port. 1 = Enable System Error generation in response to nonfatal errors reported on this port.	0b	RW
00	ESECE	**Enable system error on Correctable Error:** Controls the Root Complex's response to correctable errors reported by any of the devices in the hierarchy associated with this Root Port. 0 = Disable System Error generation in response to correctable errors reported on this port. 1 = Enable System Error generation in response to correctable errors reported on this port.	0b	RW

13.4.1.53 Offset 84 - 87h: PEARPSTS – PCI Express Root Port Status Register

This register supports power management events.

Table 352. Offset 84 - 87h: PEARPSTS – PCI Express Root Port Status Register

Device: 2	***Function:*** 0
Offset: *84 - 87h*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
31:18	Reserved	Reserved	0000h	
17	PMEP	**PME Pending:** 0 = Cleared by hardware when no more PMEs are pending. 1 = Indicates that another PME is pending when the PME Status bit is set.	0b	RO
16	PMES	**PME Status:** 0 = Cleared by software writing a '1' to the bit location. 1 = PME has been asserted by the requestor indicated in the PME Requestor ID field. ***Note:*** Subsequent PMEs are kept pending until cleared by software.	0b	RWC
15:00	PMERID	**PME Requestor ID:** Indicates the PCI Requestor ID of the last PME requestor.	0000h	RO

13.4.1.54 Offset 100 - 103h: ENHCAPST – Enhanced Capability Structure Register

This register identifies the capability structure and points to the next structure. This enhanced configuration structure by definition starts at configuration offset 100h.

Table 353. Offset 100 - 103h: ENHCAPST – Enhanced Capability Structure Register

Device: 2	***Function:*** 0
Offset: *100-103h*	***Size:*** *32 bits*
Default Value: *0001_0001h*	

Bits	Name	Description	Reset Value	Access
31:20	NCP	**Next Capability Pointer:** This field is hardwired to 000h to indicate that there are no other items in the capability list.	000h	RO
19:16	CV	**Capability Version:** Hardwired to 1h, to indicate *PCI Express* Interface Specification, Rev 1.0a.*	1h	RO
15:00	Extended CAP_ID	Hardwired to 0001h, to indicate advanced error reporting capability.	0001h	RO

13.4.1.55 Offset 104 - 107h: UNCERRSTS – Uncorrectable Error Status Register

The Uncorrectable Error Status register reports the status of individual error sources on the PCI Express device. An individual error status bit that is set indicates that a particular error occurred. Software may clear an error status bit by writing a ‘1’ to the bit location. These bits are sticky through reset.

Table 354. Offset 104 - 107h: UNCERRSTS – Uncorrectable Error Status Register (Sheet 1 of 2)

Device: 2 ***Function:*** 0
Offset: *104 - 107h* ***Size:*** *32 bits*
Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:21	Reserved	Reserved	000h	
20	USR_UNCERRSTS	**Unsupported Request [STICKY]:** This error, if the first uncorrectable error, loads the header log. This bit is sticky through reset. 0 = Cleared by writing a ‘1’ to the bit location. 1 = Unsupported Request detected.	0b	RWC
19	EES	**ECRC Error Status [STICKY]:** This error, if the first uncorrectable error, loads the header log. This bit is sticky through reset. 0 = Cleared by writing a ‘1’ to the bit location. 1 = ECRC error detected.	0b	RWC
18	MTS	**Malformed TLP Status [STICKY]:** This error, if the first uncorrectable error, loads the header log. Malformed TLP errors include: data payload length issues, byte enable rule violations, and various other illegal field settings. This bit is sticky through reset. 0 = Cleared by writing a ‘1’ to the bit location. 1 = Malformed TLP detected	0b	RWC
17	ROS	**Receiver Overflow Status [STICKY]:**Optional PCI Express specification bit, implemented for IMCH. This error, if the first uncorrectable error, loads the header log. IMCH checks for overflows on the following upstream queues: posted, non-posted, and completion. This bit is sticky through reset. 0 = Cleared by writing a ‘1’ to the bit location. 1 = Receiver Overflow detected.	0b	RWC
16	UCS	**Unexpected Completion Status [STICKY]:** This bit is set when the device receives a completion which does not correspond to any of the outstanding requests issued by that device. This error, if the first uncorrectable error, loads the header log. This bit is sticky through reset. 0 = Cleared by writing a ‘1’ to the bit location. 1 = Unexpected Completion detected.	0b	RWC
15	CAS	**Completer Abort Status [STICKY:]** Optional PCI Express specification bit, implemented for IMCH. If a request received violates the specific programming model of this device, but is otherwise legal, this bit is set. This error, if the first uncorrectable error, load the header log. This bit is sticky through reset. 0 = Cleared by writing a ‘1’ to the bit location. 1 = Completer Abort detected.	0b	RWC
14	CTS	**Completion Timeout Status [STICKY]:** The Completion Timeout timer must expire if a Request is not completed in 50 ms, but must not expire earlier than 50 µs. When the timer expires, this bit is set. This bit is sticky through reset. 0 = Cleared by writing a ‘1’ to the bit location. 1 = Completion timeout detected.	0b	RWC

Table 354. Offset 104 - 107h: UNCERRSTS – Uncorrectable Error Status Register (Sheet 2 of 2)

Device: 2	**Function:** 0
Offset: 104 - 107h	**Size:** 32 bits
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
13	FCPES	**Flow Control Protocol Error Status [STICKY]:**Optional PCI Express specification bit, implemented for IMCH. 0 = Cleared by writing a '1' to the bit location. 1 = Flow Control Protocol Error detected. Intel® 3100 Chipset asserts this bit for one of two conditions: • An FC update has been received which describes header or data credits for P, NP, or CPL which were originally advertised as infinite during initialization but are now advertised with non-zero or non-infinite values. • The number of credits advertised in an update is less than the number of credits in the previous update. The hardware accepts this flow control update, as it cannot determine if this update or the previous one was in error. This bit is sticky through reset.	0b	RWC
12	PTS	**Poisoned TLP Status [STICKY]:** This bit when set indicates that some portion of the TLP data payload was corrupt. This error, if the first uncorrectable error, loads the header log. This bit is sticky through reset. 0 = Cleared by writing a '1' to the bit location. 1 = Poisoned TLP detected.	0b	RWC
11:05	Reserved	Reserved	00h	
04	DLPES	**Data Link Protocol Error Status [STICKY]:** This bit is set when an ACK/NAK received does not specify the sequence number of an unacknowledged TLP, or of the most recently acknowledged TLP. This bit is sticky through reset. 0 = Cleared by writing a '1' to the bit location. 1 = Data Link Protocol Error detected.	0b	RWC
03:00	Reserved	Reserved	000b	

13.4.1.56 Offset 108 - 10Bh: UNCERRMSK – Uncorrectable Error Mask Register

The Uncorrectable Error Mask register controls reporting of individual errors by device to the PCI Express Root Complex via a PCI Express error message. A masked error (respective bit set in mask register) is not reported to the PCI Express Root Complex by an individual device. However, masked errors are still logged in the Uncorrectable Error Status register. There is one mask bit corresponding to every implemented bit in the Uncorrectable Error Status register. These bits are sticky through reset.

Table 355. Offset 108 - 10Bh: UNCERRMSK – Uncorrectable Error Mask Register

Device: *2* ***Function:*** *0*
Offset: *108 - 10Bh* ***Size:*** *32 bit*
Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:21	Reserved	Reserved	000h	
20	USR_UNCERRMSK	**Unsupported Request[STICKY]:** 0 = Report Unsupported Request Error 1 = Mask Unsupported Request Error	0b	RW
19	EEM	**ECRC Error Mask[STICKY]:** 0 = Report ECRC Error 1 = Mask ECRC Error	0b	RW
18	MTM	**Malformed TLP Mask[STICKY]:** 0 = Report Malformed TLP Error 1 = Mask Malformed TLP Error	0b	RW
17	ROM	**Receiver Overflow Mask[STICKY]:** Optional PCI Express specification bit, implemented for IMCH. 0 = Report Receiver Overflow Error 1 = Mask Receiver Overflow Error	0b	RW
16	UCM	**Unexpected Completion Mask[STICKY]:** 0 = Report Receiver Overflow Error 1 = Mask Receiver Overflow Error	0b	RW
15	CAM	**Completer Abort Mask[STICKY]:** Optional PCI Express specification bit, implemented for IMCH. 0 = Report Completer Abort Error 1 = Mask Completer Abort Error	0b	RW
14	CTM	**Completion Timeout Mask[STICKY]:** 0 = Report Completion Timeout Error 1 = Mask Completion Timeout Error	0b	RW
13	FCPEM	**Flow Control Protocol Error Mask[STICKY]:** Optional PCI Express specification bit, implemented for IMCH. 0 = Report Flow Control Protocol Error 1 = Mask Flow Control Protocol Error	0b	RW
12	PTM	**Poisoned TLP Mask[STICKY]:** 0 = Report Poisoned TLP Error 1 = Mask Poisoned TLP Error	0b	RW
11:05	Reserved	Reserved	00h	
04	DLPEM	**Data Link Protocol Error Mask[STICKY]:** 0 = Report Data Link Protocol Error 1 = Mask Data Link Protocol Error	0b	RW
03:00	Reserved	Reserved	000b	

13.4.1.57 Offset 10C - 10Fh: UNCERRSEV – Uncorrectable Error Severity Register

The Uncorrectable Error Severity register controls whether an individual error is reported as a nonfatal or fatal error. An error is reported as fatal when the corresponding error bit in the severity register is set. If the bit is cleared, the corresponding error is considered nonfatal. These bits are sticky through reset.

Table 356. Offset 10C - 10Fh: UNCERRSEV – Uncorrectable Error Severity Register

Device: *2*	***Function:*** *0*
Offset: *10C - 10Fh*	***Size:*** *32 bits*
Default Value: *0006_2010h*	

Bits	Name	Description	Reset Value	Access
31:21	Reserved	Reserved	000h	
20	USR_UNCERRSEV	**Unsupported Request[STICKY]:** 0 = Nonfatal 1 = Fatal	0b	RW
19	EESEV	**ECRC Error Severity[STICKY]:** 0 = Nonfatal 1 = Fatall	0b	RW
18	MTSEV	**Malformed TLP Severity[STICKY]:** 0 = Nonfatal 1 = Fatal	1b	RW
17	ROSEV	**Receiver Overflow Severity[STICKY]:** Optional PCI Express specification bit, implemented for IMCH. 0 = Nonfatal 1 = Fatal	1b	RW
16	UCSEV	**Unexpected Completion Severity[STICKY]:** 0 = Nonfatal 1 = Fatal	0b	RW
15	CASEV	**Completer Abort Severity [STICKY]:** Optional PCI Express specification bit, implemented for IMCH. 0 = Nonfatal 1 = Fatal	0b	RW
14	CTSEV	**Completion Timeout Severity[STICKY]:** 0 = Nonfatal 1 = Fatal	0b	RW
13	FCPESEV	**Flow Control Protocol Error Severity[STICKY]:** Optional PCI Express specification bit, implemented for IMCH. 0 = Nonfatal 1 = Fatal	1b	RW
12	PTSEV	**Poisoned TLP Severity[STICKY]:** 0 = Nonfatal 1 = Fatal	0b	RW
11:05	Reserved	Reserved	00h	
04	DLPESEV	**Data Link Protocol Error Severity[STICKY]:** 0 = Nonfatal 1 = Fatal	1b	RW
03:00	Reserved	Reserved	0000b	

13.4.1.58 Offset 110 - 113h: CORERRSTS – Correctable Error Status Register

The Correctable Error Status register reports the status of individual error sources on the PCI Express device. An individual error status bit that is set indicates that a particular error occurred. Software may clear an error status bit by writing a ‘1’ to the bit location. These bits are sticky through reset.

Table 357. Offset 110 - 113h: CORERRSTS – Correctable Error Status Register

Device: 2 ***Function:*** 0
Offset: *110 - 113h* ***Size:*** *32 bit*
Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:13	Reserved	Reserved	0_0000h	
12	RTTS	**Replay Timer Timeout Status[STICKY]:** The replay timer counts time since the last ACK or NAK DLLP was received. When the timer expires, this bit is set. This bit is sticky through system reset. 0 = Cleared by writing a ‘1’ to the bit location. 1 = Replay Timer timeout detected.	0b	RWC
11:09	Reserved	Reserved	000b	
08	RNRS	**REPLAY_NUM Rollover Status[STICKY]:** A 2-bit counter counts the number of times the retry buffer has been retransmitted. When this counter rolls over, this bit is set. This bit is sticky through system reset. 0 = Cleared by writing a ‘1’ to the bit location. 1 = REPLAY_NUM rollover detected.	0b	RWC
07	BDS	**Bad DLLP Status[STICKY]:** This bit is set when the calculated DLLP CRC is not equal to the received value. This bit is sticky through system reset. 0 = Cleared by writing a ‘1’ to the bit location. 1 = Bad DLLP detected.	0b	RWC
06	BTS	**Bad TLP Status[STICKY]:** This bit is set when the calculated TLP CRC is not equal to the received value. Also included are invalid sequence numbers. 0 = Good TLP status. 1 = Bad TLP status.	0b	RWC
05:01	Reserved	Reserved	00h	
00	RES	**Receiver Error Status[STICKY]:** Optional PCI Express specification bit, implemented for IMCH. Data is delivered over PCI Express via packets built out of 8b/10b symbols. Receiver Error Status register is set for 8b/10b errors received, framing errors received irrespective of the packet boundaries. 0 = Cleared by writing a ‘1’ to the bit location. 1 = Receiver Error detected. ***Note:***	0b	RWC

13.4.1.59 Offset 114 - 117h: CORERRMSK – Correctable Error Mask Register

The Correctable Error Mask register controls reporting of individual errors by device to the PCI Express Root Complex via a PCI Express error message. A masked error (respective bit set in mask register) is not reported to the PCI Express Root Complex by an individual device. However, masked errors are still logged in the Correctable Error Status register. There is one mask bit corresponding to every implemented bit in the Correctable Error Status register. These bits are sticky through reset.

Table 358. Offset 114 - 117h: CORERRMSK – Correctable Error Mask Register

Device: 2 ***Function:*** 0
Offset: *114 - 117h* ***Size:*** *32 bit*
Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:13	Reserved	Reserved	0_0000h	
12	RTTM	**Replay Timer Timeout Mask[STICKY]:** This bit is sticky through system reset. 0 = Report Replay Timer Timeout error. 1 = Mask Replay Timer timeout error.	0b	RW
11:09	Reserved	Reserved	000b	
08	RNRM	**REPLAY_NUM Rollover Mask[STICKY]:** This bit is sticky through system reset. 0 = Report REPLAY_NUM rollover 1 = Mask REPLAY_NUM rollover.	0b	RW
07	BDM	**Bad DLLP Mask[STICKY]:** This bit is sticky through system reset. 0 = Report Bad DLLP error. 1 = Mask Bad DLLP error.	0b	RW
06	BTM	**Bad TLP Mask[STICKY]:** This bit is sticky through system reset. 0 = Report Bad TLP error. 1 = Mask Bad TLP error.	0b	RW
05:01	Reserved	Reserved	00h	
00	REM	**Receiver Error Mask[STICKY]:** This bit is sticky through system reset. 0 = Report Receiver error. 1 = Mask Receiver error.	0b	RW

13.4.1.60 Offset 118 - 11Bh: AERCACR – Advanced Error Capabilities and Control Register

This register identifies the capability structure and points to the next structure. The first error pointer rearms after the unmasked errors have been cleared. Software, after clearing the errors, must read the register again to ensure that it is indeed cleared. If it finds that another error occurred, it can not rely on the pointer or header, unless it detects that the error pointer changed from the last time it was read for the previous error. Bits in this register also declare the ECRC capability of this device. Some of these bits are sticky through reset as indicated below. Hardware is required to gracefully switch between enabled/disabled for both ECRC generate and check, although it is software's responsibility to properly handle the change in behavior of any TLPs in flight.

Table 359. Offset 118 - 11Bh: AERCACR – Advanced Error Capabilities and Control Register

Device: 2	**Function:** 0
Offset: 118 - 11Bh	**Size:** 32 bit
Default Value: 0000_00A0h	

Bits	Name	Description	Reset Value	Access
31:09	Reserved	Reserved	00_0000h	
08	ECE	**ECRC Check Enable [STICKY]:** This bit is sticky through system reset. 0 = Disable ECRC check. 1 = Enable ECRC check.	0b	RW
07	EC	**ECRC Check Capable:** This bit reflects if the device is capable of checking ECRC. 0 = Not capable of checking ECRC. 1 = Capable of checking ECRC.	1b	RO
06	EGE	**ECRC Generation Enable [STICKY]:** This bit is sticky through system reset. 0 = Disable ECRC generation. 1 = Enable ECRC generation.	0b	RW
05	EGC	**ECRC Generation Capable:** This bit reflects if the device is capable of checking ECRC. 0 = Not capable of generating ECRC. 1 = Capable of generating ECRC.	1b	RO
04:00	FEP	**First error pointer [STICKY]:** Identifies the bit position of the first error reported in the Uncorrectable Error Status register. However, if a subsequent Uncorrectable Error occurs with a higher severity, this field is over-written with the bit position of the subsequent error status bit. Also, if multiple errors of equal severity are logged simultaneously, this field identifies the bit position of the most significant (leftmost) bit that has been set in the Uncorrectable Error Status register. In the event of simultaneous errors, the pointer indicates the least significant bit of the group. This field is sticky through system reset.	00000b	RO

13.4.1.61 Offset 11C - 11Fh: HDRLOG0 – Header Log DW 0 (1st 32 bits) Register

This register contains the first 32 bits of the header log locked down when the first uncorrectable error occurs that saves the header. To rearm this register all reported uncorrectable errors must be cleared from the register. Software after clearing the errors must read the register again to ensure that it is indeed cleared. If it finds that another error occurred, it can not rely on the pointer or header, unless it detects that the error pointer changed from the last time it was read for the previous error. Byte 0 of the header is located in byte 3 of the Header Log Register 0, byte 1 of the header is in byte 2 of the Header Log Register 0 and so forth. For 12 byte headers, only the first three of the four Header Log Registers are used, and values in HDRLOG3 are undefined. These bits are sticky through reset.

Table 360. Offset 11C - 11Fh: HDRLOG0 – Header Log DW 0 (1st 32 bits) Register

Device: 2	**Function:** 0
Offset: 11C - 11Fh	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
31:00	HL0	**Header Log 0[STICKY]:**A masked error (respective bit set to '1' in mask register) is not logged in the Header Log Register, does not update the First Error Pointer, and is not reported to the PCI Express Root Complex by an individual device.	0000_0000h	RO

13.4.1.62 Offset 120 - 123h: HDRLOG1 – Header Log DW 1 (2nd 32 bits) Register

The function of the Header Log registers is described in Section 13.4.1.61 on page -457. Header Log DW1 contains the second 32 bits of the header. Byte 4 of the header is located in byte 3 of the Header Log Register 1, byte 5 of the header is in byte 2 of the Header Log Register 1 and so forth. These bits are sticky through reset.

Table 361. Offset 120 - 123h: HDRLOG1 – Header Log DW 1 (2nd 32 bits) Register

Device: 2	**Function:** 0
Offset: 120 - 123h	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
31:00	HL1	**Header Log 1[STICKY]:** A masked error (respective bit set to '1' in mask register) is not logged in the Header Log Register, does not update the First Error Pointer, and is not reported to the PCI Express Root Complex by an individual device. These bits are sticky through system reset.	0000_0000h	RO

13.4.1.63 Offset 124 - 127h: HDRLOG2 – Header Log DW 2 (3rd 32 bits) Register

The function of the Header Log registers is described in Section 13.4.1.61 on page -457. Header Log DW2 contains the third 32 bits of the header. Byte 8 of the header is located in byte 3 of the Header Log Register 2, byte 9 of the header is in byte 2 of the Header Log Register 2 and so forth. These bits are sticky through reset.

Table 362. Offset 124 - 127h: HDRLOG2 – Header Log DW 2 (3rd 32 bits) Register

Device: 2	**Function:** 0
Offset: 124 - 127h	**Size:** 32 bits
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
31:00	HL2	**Header Log 2[STICKY]:** A masked error (respective bit set to '1' in mask register) is not logged in the Header Log Register, does not update the First Error Pointer, and is not reported to the PCI Express Root Complex by an individual device. These bits are sticky through system reset.	0000_0000h	RO

13.4.1.64 Offset 128 - 12Bh: HDRLOG3 – Header Log DW 3 (4th 32 bits) Register

The function of the Header Log registers is described in Section 13.4.1.61 on page -457. Header Log DW3 contains the fourth 32 bits of the header. For 16-byte headers, byte 12 of the header is located in byte 3 of the Header Log Register 3, byte 13 of the header is in byte 2 of the Header Log Register 3 and so forth. For 12 byte headers, values in this register are undefined. These bits are sticky through reset.

Table 363. Offset 128 - 12Bh: HDRLOG3 – Header Log DW 3 (4th 32 bits) Register

Device: 2 ***Function:*** 0
Offset: *128 - 12Bh* ***Size:*** *32 bit*
Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:00	HL3	**Header Log 3[STICKY]:** A masked error (respective bit set to '1' in mask register) is not logged in the Header Log Register, does not update the First Error Pointer, and is not reported to the PCI Express Root Complex by an individual device. These bits are sticky through system reset.	0000_0000h	RO

13.4.1.65 Offset 12C - 12Fh: RPERRCMD – Root (Port) Error Command Register

This register controls the generation of interrupts (beyond the basic root complex capability to generate system errors) upon detection of errors. System error generation in response to PCI Express error messages may be turned off by system software using the PCI Express Capability Structure when advanced error reporting via interrupts is enabled.

Table 364. Offset 12C - 12Fh: RPERRCMD – Root (Port) Error Command Register

Device: 2 ***Function:*** 0
Offset: *12C - 12Fh* ***Size:*** *32 bit*
Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:03	Reserved	Reserved	000_0000h	
02	FEIE	**Fatal Error Interrupt Enable:** Enables the generation of an interrupt when a fatal error is reported by any of the devices in the hierarchy associated with this Root Port. 0 = Disable interrupt generation on fatal error. 1 = Enable interrupt generation on fatal error.	0b	RW
01	NEIE	**Nonfatal Error Interrupt Enable:** Enables the generation of an interrupt when an nonfatal error is reported by any of the devices in the hierarchy associated with this Root Port. 0 = Disable interrupt generation on nonfatal error. 1 = Enable interrupt generation on nonfatal error.	0b	RW
00	CEIE	**Correctable Error Interrupt Enable:** Enables the generation of an interrupt when a correctable error is reported by any of the devices in the hierarchy associated with this Root Port. 0 = Disable interrupt generation on correctable error. 1 = Enable interrupt generation on correctable error.	0b	RW

13.4.1.66 Offset 130 - 133h: RPERRMSTS – Root (Port) Error Message Status Register

This register reports the status of errors received by the root complex. Each correctable and uncorrectable (nonfatal and fatal) error source has a First Error bit and a Next Error bit. When an error is received by the root complex, the associated First Error bit is set and the Requestor ID is logged in the Error Source Identification register. Software may clear an error status bit by writing a '1' to the bit location. If software does not clear the first reported error before another error is received, the Next Error status bit is set, but the Requestor ID of the subsequent error message is discarded. These bits are sticky through reset.

Table 365. Offset 130 - 133h: RPERRMSTS – Root (Port) Error Message Status Register (Sheet 1 of 2)

Device: 2	***Function:*** 0
Offset: *130 - 133h*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
31:27	AEIMN	**Advanced Error Interrupt Message Number:** If this function has been allocated more than one MSI interrupt number, this field reflects the offset between the base Message Data and the MSI Message that is generated when any of the status bits of this capability are set.	0h	RO
26:07	Reserved	Reserved	00_0000h	
06	FEMD	**Fatal Error Messages Detected[STICKY]:** This bit is used by error handling software to determine whether fatal errors are outstanding in the hierarchy. In hardware, this bit along with bits 4 and 2 is used to clear fatal error escalation. This bit is sticky through system reset. 0 = Software clears this bit by writing a '1' to the bit location. 1 = Fatal error message detected.	0b	RWC
05	NFEMD	**Non-Fatal Error Messages Detected[STICKY]:** This bit is used by error handling software to determine whether non-fatal errors are outstanding in the hierarchy. In hardware, this bit along with bits 4 and 2 is used to clear non-fatal error escalation. This bit is sticky through system reset. 0 = Software clears this bit by writing a '1' to the bit location. 1 = Non-fatal error message detected.	0b	RWC
04	FUFF	**First Uncorrectable Fatal Flag[STICKY]:** This bit captures the nature of the first uncorrectable error message detected (and logged in the error source ID register). This bit is sticky through system reset. 0 = First uncorrectable error is non-fatal. 1 = First uncorrectable error is fatal. Software uses this flag to determine whether the uncorrectable error source ID belongs to the fatal or non-fatal error handler routine in the event that the two are independent.	0b	RWC
03	MUEMD	**Multiple Uncorrectable Error Messages Detected [STICKY]:** In the unlikely event of two first errors occurring during the same clock period, only the first uncorrectable error message bit is set. It takes an error to occur in a subsequent clock to set this bit. These bits are sticky through system reset. 0 = Software clears this bit by writing a '1' to the bit location. 1 = Set when either a fatal or nonfatal error is received, and the First Uncorrectable Error Detected bit is already set. This indicates that one or more message Requestor IDs were lost.	0b	RWC

Table 365. Offset 130 - 133h: RPERRMSTS – Root (Port) Error Message Status Register (Sheet 2 of 2)

Device: 2	**Function:** 0
Offset: 130 - 133h	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
02	FUEMD	**First Uncorrectable Error Message Detected[STICKY]:**This bit is sticky through system reset. 0 = Software clears this bit by writing a '1' to the bit location. 1 = Set when the first fatal or nonfatal error is received.	0b	RWC
01	MCEMD	**Multiple Correctable Error Messages Detected[STICKY]:** In the unlikely event of two first errors occurring during the same clock period, only the first correctable error message bit is set. It takes an error to occur in a subsequent clock to set this bit. This bit is sticky through system reset. 0 = Software clears this bit by writing a '1' to the bit location. 1 = Set when a correctable error is received, and the First Correctable Error Detected bit is already set. This indicates that one or more message Requestor IDs were lost.	0b	RWC
00	FCEMD	**First Correctable Error Message Detected[STICKY]:** The Root Error Status bit reports status of error messages (ERR_COR) received by the root complex, and of errors detected/reported (not masked) by the Root Port itself. This bit is sticky through system reset. 0 = Software clears this bit by writing a '1' to the bit location. 1 = Set when the first correctable error is received.	0b	RWC

13.4.1.67 Offset 134 - 137h: ERRSID – Error Source ID Register

This register reports the source (Requestor ID) of the first correctable and uncorrectable (fatal or nonfatal) errors reported in the Root Error Status register. This register is updated regardless of the settings of the Root Control register and the Root Error Command register. These bits are sticky through reset.

Table 366. Offset 134 - 137h: ERRSID – Error Source ID Register

Device: 2	**Function:** 0
Offset: 134 - 137h	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
31:16	UESID	**Uncorrectable Error Source ID[STICKY]:** Requestor ID of the source when an uncorrectable error (fatal or nonfatal) is received, and the First Uncorrectable Error Detected bit is not already set. Since this ID could be for an internally detected error or from a message received from the other end of the link, in the event of errors detected in the same clock, priority is given to the error received from the link, and that ID is what is logged. These bits are sticky through system reset.	0000h	RO
15:00	CESID	**Correctable Error Source ID [STICKY]:**Requestor ID of the source when an correctable error is received, and the First Correctable Error Detected bit is not already set. Since this ID could be for an internally detected error or from a message received from the other end of the link, in the event of errors detected in the same clock, priority is given to the error received from the link, and that ID is what is logged. These bits are sticky through system reset.	0000h	RO

13.4.1.68 Offset 140 - 143h: PEAUNITERR – PCI Express Unit Error Register

This register is specific to the IMCH. It captures the non-PCI Express unit errors (those beyond the scope of the bus specification). The unit error mechanism is parallel to that used by "compatible" error registers and masks, but cannot feed back into standard registers because that would confuse standardized error handling software (which would not understand the extracurricular error bits). Escalation is controlled via the PEAERRDOCMD register (Table 369, "Offset 148 - 14Bh: PEAERRDOCMD – PCI Express Error Do Command Register" on page 466) for both standard and IMCH-specific error types. Uncorrectable fatal errors feed into the fatal reporting select, uncorrectable non-fatal errors feed into the non-fatal reporting select, and correctable errors feed into the correctable reporting select. The lower nibble is for HPC related errors. These bits are sticky through reset.

Table 367. Offset 140 - 143h: PEAUNITERR – PCI Express Unit Error Register (Sheet 1 of 3)

Device: *2* ***Function:*** *0*
Offset: *140 - 143h* ***Size:*** *32 bit*
Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved for future additions.	0000h	
15	UPQOS	**Upstream Posted Queue Overflow Status[STICKY]:** This bit is one of the components of the Receiver Overflow Status bit in the UNCERRSTS register. Even though this bit can be set, it is only reported through the receiver overflow bit in the UNCERRSTS register. The setting of this bit is never logged in the local FERR/NERR registers or subsequently the global FERR/NERR registers, nor does it cause a SCI/SMI/SERR or MCERR message. At most, when the report mask is disabled, it could affect the unit error pointer. This functionality is provided as an aid to debug. 0 = Software clears this bit by writing a '1' to the bit position. 1 = Overflow occurred for one of the posted header or data queues.	0b	RWC
14	UNPQOS	**Upstream Non-Posted Queue Overflow Status[STICKY]:** This bit is set if an overflow occurs for the non-posted header queue. There is no upstream non-posted data queue. It is one of the components of the Receiver Overflow Status bit in the UNCERRSTS register. Even though this bit can be set, it is only reported through the receiver overflow bit in the UNCERRSTS register. The setting of this bit is never logged in the local FERR/NERR registers or subsequently the global FERR/NERR registers, nor is it a cause for a SCI/SMI/SERR or MCERR message. At most, when the report mask is disabled, it could affect the unit error pointer. This functionality is provided as an aid to debug. 0 = Software clears this bit by writing a '1' to the bit position. 1 = Overflow occurred for the non-posted header queues.	0b	RWC
13	UCQOS	**Upstream Completion Queue Overflow Status [STICKY]:** This bit is set if an overflow occurs for either the completion header or data queues It is one of the components of the Receiver Overflow Status bit in the UNCERRSTS register. Even though this bit can be set, it is only reported through the receiver overflow bit in the UNCERRSTS register. The setting of this bit is never logged in the local FERR/NERR registers or subsequently the global FERR/NERR registers, nor is it a cause a SCI/SMI/SERR or MCERR message. This functionality is provided as an aid to debug. 0 = Software clears this bit by writing a '1' to the bit position. 1 = Overflow occurred for one of the completion header or data queues.	0b	RWC

Table 367. Offset 140 - 143h: PEAUNITERR – PCI Express Unit Error Register (Sheet 2 of 3)

Device: 2 **Function:** 0
Offset: 140 - 143h **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
12	LPE	**LLE Protocol Error [STICKY]:** This bit is set when the transaction layer detects a protocol error on the receiver interface from the LLE. Such an event should cause retraining eventually, but not necessarily immediately. The transaction with a problem is dropped. 0 = Software clears this bit by writing a '1' to the bit position. 1 = Transaction layer detected a protocol error on the receiver interface from the LLE	0b	RWC
11	LDE	**Link Down Error [STICKY]:** 0 = Software clears this bit by writing a '1' to the bit position. 1 = Set when the link transitions from DL_UP to DL_DOWN.	0b	RWC
10	DDQPE	**Downstream Data Queue Parity Error [STICKY]:** A parity error occurred in the downstream data queue. 0 = Software clears this bit by writing a '1' to the bit position. 1 = Parity error occurred in the downstream data queue.	0b	RWC
09	LPELT	**LUT Parity Error Parity error occurred in the Lookup Table [STICKY]:** The logic is now lost and no transfers can be processed. The link is forced into retraining. (When this error occurs, it is most likely to be reported as multiple errors, since the error is reported as long as the row with the error is accessed.) 0 = Software clears this bit by writing a '1' to the bit position. 1 = Parity error occurred in the Lookup Table.	0b	RWC
08	LDPE	**LLRB Data Parity Error [STICKY]:** Data Parity error occurred in the Link Level Retry Buffer. This error can only occur during a retry. 0 = Software clears this bit by writing a '1' to the bit position. 1 = Data Parity error occurred in the Link Level Retry Buffer.	0b	RWC
07	LHPE	**LLRB Header Parity Error [STICKY]:** The current transaction is terminated as marked as bad. No further transfers are completed. This error is really Uncorrectable, but this event forces retraining which by definition makes it fatal. This error can only occur during a retry. 0 = Software clears this bit by writing a '1' to the bit position. 1 = LLRB header parity error detected.	0b	RWC
06	LCPE	**LLRB Control Parity Error [STICKY]:** Since the pointer information is somehow lost or corrupted, no further data transfers can be completed. This error can only occurring during a retry. 0 = Software clears this bit by writing a '1' to the bit position. 1 = LLRB control parity error detected.	0b	RWC
05:03	DLLPTOE	**DLLP Timeout Error [Sticky]:** Set when DLLP traffic is not received within the expected time. Only asserted when flow control traffic is not received within the expected time. This bit is not set for timeouts related to ACK or NAK. This bit is correctable.	000b	RWC

Table 367. Offset 140 - 143h: PEAUNITERR – PCI Express Unit Error Register (Sheet 3 of 3)

Device: 2	**Function:** 0
Offset: 140 - 143h	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
02	SMBCLTO	**SMB Clock Low Timeout [STICKY]:** 0 = Software clears this bit by writing a '1' to the bit position. 1 = SMB CLK low greater than 25 ms.	0b	RWC
01	UESMBN	**Unexpected NAK on SMB [STICKY]**: 0 = Software clears this bit by writing a '1' to the bit position. 1 = Unexpected NAK on SMB detected.	0b	RWC
00	SMBLA	**SMB Lost Bus Arbitration** (Correctable) **[STICKY]:** This bit is sticky through reset. 0 = Software clears this bit by writing a '1' to the bit position. 1 = SMB lost bus arbitration.	0b	RWC

13.4.1.69 Offset 144 - 147h: PEAMASKERR – PCI Express Unit Mask Error Register

This register is used for selecting the global error reporting method for the various error conditions.

Table 368. Offset 144 - 147h: PEAMASKERR – PCI Express Unit Mask Error Register (Sheet 1 of 2)

Device: 2	**Function:** 0
Offset: 144 - 147h	**Size:** 32 bit
Default Value: 0000_E000h	

Bits	Name	Description	Reset Value	Access
31:16	Reserved	For future additions.	0000h	
15	UPQOM	**Upstream Posted Queue Overflow Mask[STICKY]:** Defaults to masked, normally reported through PCI Express receive overflow status bit. 0 = Enable upstream posted queue overflow reporting. 1 = Disable upstream posted queue overflow reporting.	1b	RW
14	UNPQOM	**Upstream Non-Posted Queue Overflow Mask[STICKY]:** Defaults to masked, normally reported through PCI Express receive overflow status bit. 0 = Enable upstream non-posted queue overflow reporting. 1 = Disable upstream non-posted queue overflow reporting.	1b	RW
13	UCQOM	**Upstream Completion Queue Overflow Mask[STICKY]:** Defaults to masked, normally reported through PCI Express receive overflow status bit. 0 = Enable upstream completion queue overflow reporting. 1 = Disable upstream completion queue overflow reporting.	1b	RW
12	LPEM	**LLE Protocol Error Mask[STICKY]:** 0 = Enable LLE protocol error reporting. 1 = Disable LLE protocol error reporting.	0b	RW
11	LDEM	**Link Down Error Mask[STICKY]:** Mask reporting of detected link transitions from DL_UP to DL_DOWN. 0 = Enable link down error mask reporting. 1 = Disable link down error mask reporting.	0b	RW

Table 368. Offset 144 - 147h: PEAMASKERR – PCI Express Unit Mask Error Register (Sheet 2 of 2)

Device: 2	**Function:** 0
Offset: 144 - 147h	**Size:** 32 bit
Default Value: 0000_E000h	

Bits	Name	Description	Reset Value	Access
10	DDQPERM	**Downstream Data Queue Parity Error Reporting Mask[STICKY]:** 0 = Enable Downstream Data Queue Parity Error Reporting. 1 = Disable Downstream Data Queue Parity Error Reporting.	0b	RW
09	LPERM	**LUT Parity Error Reporting Mask[STICKY]:** 0 = Enable LUT parity error reporting. 1 = Disable LUT parity error reporting.	0b	RW
08	LDPERM	**LLRB Data Parity Error Reporting Mask[STICKY]:** 0 = Enable LLRB data parity error reporting. 1 = Disable LLRB data parity error reporting.	0b	RW
07	LHPERM	**LLRB Header Parity Error Reporting Mask[STICKY]:** 0 = Enable LLRB data parity error reporting. 1 = Disable LLRB data parity error reporting.	0b	RW
06	LCPERM	**LLRB Control Parity Error Reporting Mask[STICKY]:** 0 = Enable LLRB control parity error reporting. 1 = Disable LLRB control parity error reporting.	0b	RW
05	DTERM	**DLLP Timeout Error Reporting Mask[STICKY]:** 0 = Enable DLLP Timeout error reporting. 1 = Disable DLLP Timeout error reporting.	00b	RW
04:03	Reserved	Reserved	00b	
02	SMBCLTORM	**SMBCLTO Reporting Mask [STICKY]:** 0 = Enable SMBCLTO reporting. 1 = Disable SMBCLTO reporting.	0b	RW
01	UESMBNRM	**UESMBN Reporting Mask [STICKY]:** 0 = Enable UESMBN reporting. 1 = Disable UESMBN reporting.	0b	RW
00	SMBLARM	**SMBLA Reporting Mask [STICKY]:** 0 = Enable SMBLA reporting. 1 = Disable SMBLA reporting.	0b	RW

13.4.1.70 Offset 148 - 14Bh: PEAERRDOCMD – PCI Express Error Do Command Register

This register supports PCI Express error commands for doing various signaling. DO_SCI, DO_SMI, and DO_MCERR, DO_SERR must further be enabled by the PCI Express Host Do Command register.

Table 369. Offset 148 - 14Bh: PEAERRDOCMD – PCI Express Error Do Command Register (Sheet 1 of 2)

Device: 2 **Function:** 0
Offset: 148 - 14Bh **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:29	Reserved	Reserved	000b	
28:24	FEPUPCE	**First Error Pointer for unmasked PCI Express correctable errors [STICKY]:**This pointer is rearmed when all unmasked errors have been cleared. In the event of simultaneous errors, the pointer indicates the least significant bit of the group. These bits are sticky.	00h	RO
23:21	Reserved	Reserved	000b	
20:16	FEPPE	**First Error Pointer for PCI Express-unit errors[STICKY]:** This pointer is locked once any units errors are logged in the PEAFERR. It is rearmed when all PEAUNIT errors have been cleared. In the event of simultaneous errors, the pointer indicates the least significant bit of the group. This pointer is only valid for an error that is enabled for reporting. These bits are sticky.	00h	RO
15	EHLULPE	**Enable Header Log use for LLE Protocol Error:** The header log is used by PCI Express uncorrectable errors. This feature is used to capture the header log for the LLE protocol error in the unit error register during debug. 0 = Disable 1 = Enable	0b	RW
14	PURE	**PCI Express Unit report enable:** This bit enables reporting of fatal or non-fatal or correctable unit errors. 0 = Disable 1 = Enable	0b	RW
13:12	PURSFE	PCI Express **unit report steering for fatal errors:** 00b = SCI 10b = SERR 01b = SMI 11b = MCERR	00b	RW
11:10	PURSNFE	**PCI Express unit report steering for non-fatal errors:** 00b = SCI 10b = SERR 01b = SMI 11b = MCERR	00b	RW
09:08	PURSCE	**PCI Express unit report steering for correctable errors:** 00b = SCI 10b = SERR 01b = SMI 11b = MCERR	00b	RW
07:06	Reserved	Reserved	00b	

Table 369. Offset 148 - 14Bh: PEAERRDOCMD – PCI Express Error Do Command Register (Sheet 2 of 2)

Device: 2	**Function:** 0
Offset: 148 - 14Bh	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
05:04	RPRSFE	**Root Port report steering for fatal errors:** If the System Error on Fatal Error bit in the Root Port Control register is set, all fatal root port errors are reported via SERR regardless of the setting of this register. MSI Enable takes precedence for this capability feature. 00b = SCI 10b = SERR 01b = SMI 11b = MCERR	00b	RW
03:02	RPRSNFE	**Root Port report steering for non-fatal errors:** If the System Error on Nonfatal Error bit in the Root Port Control register is set, all nonfatal root port errors are reported via SERR regardless of the setting of this register. MSI Enable takes precedence for this capability feature. 00b = SCI 10b = SERR 01b = SMI 11b = MCERR	00b	RW
01:00	RPRSCE	**Root Port report steering for correctable errors:** If the System Error on Correctable Error bit in the Root Port Control register is set, all correctable root port errors are reported via SERR regardless of the setting of this register. Note that MSI Enable takes precedence for this capability feature. 00b = SCI 10b = SERR 01b = SMI 11b = MCERR	00b	RW

13.4.1.71 Offset 14C - 14Fh: UNCEDMASK – Uncorrectable Error Detect Mask Register

The Uncorrectable Error Detect Mask register controls detection of the individual errors. An error event that is masked in this register is treated as though the error never happened, and is subsequently not logged in the Uncorrectable Error Status register, nor is it ever reported. There is one mask bit corresponding to every implemented bit in the Uncorrectable Error Status register. These bits are sticky through reset.

Table 370. Offset 14C - 14Fh: UNCEDMASK – Uncorrectable Error Detect Mask Register

Device: 2 **Function:** 0
Offset: 14C - 14Fh **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:21	Reserved	Reserved	000h	
20	UREDM	**Unsupported Request Error Detect Mask. [STICKY]:** 0 = Detect Unsupported Request Error 1 = Disable Unsupported Request Error detection	0b	RW
19	EEDM	**ECRC Error Detect Mask. [STICKY]:** OPTIONAL 0 = Detect ECRC Error 1 = Disable ECRC Error detection	0b	RW
18	MTEDM	**Malformed TLP Error Detect Mask. [STICKY]:** 0 = Detect Malformed TLP Error 1 = Disable Malformed TLP Error detection	0b	RW
17	ROEDM	**Receiver Overflow Error Detect Mask. [STICKY]:** OPTIONAL 0 = Detect Receiver Overflow Error 1 = Disable Receiver Overflow Error detection	0b	RW
16	UCEDM	**Unexpected Completion Error Detect Mask. [STICKY]:** 0 = Detect Unexpected Completion Error 1 = Disable Unexpected Completion Error detection	0b	RW
15	CAEDM	**Completer Abort Error Detect Mask. [STICKY]:** OPTIONAL 0 = Detect Completer Abort Error 1 = Disable Completer Abort Error detection	0b	RW
14	CTEDM	**Completion Timeout Error Detect Mask. [STICKY]:** 0 = Detect Completion Timeout Error 1 = Disable Completion Timeout Error detection	0b	RW
13	FCPEDM	**Flow Control Protocol Error Detect Mask. [STICKY]:** OPTIONAL 0 = Detect Flow Control Protocol Error 1 = Disable Flow Control Protocol Error detection	0b	RW
12	PTEDM	**Poisoned TLP Error Detect Mask. [STICKY]:** 0 = Detect Poisoned TLP Error 1 = Disable Poisoned TLP Error detection	0b	RW
11:05	Reserved	Reserved	00h	
04	DLPEDM	**Data Link Protocol Error Detect Mask. [STICKY]:** 0 = Detect Data Link Protocol Error 1 = Disable Data Link Protocol Error detection	0b	RW
03:00	Reserved	Reserved	0000b	

13.4.1.72 Offset 150 - 153h: COREDMASK – Correctable Error Detect Mask Register

The Correctable Error Detect Mask register controls detection of the individual errors. An error event that is masked in this register is not logged in the Correctable Error Status register, and is never reported. There is one mask bit corresponding to every implemented bit in the Correctable Error Status register. These bits are sticky through reset.

Table 371. Offset 150 - 153h: COREDMASK – Correctable Error Detect Mask Register

Device: 2 **Function:** 0
Offset: 150 - 153h **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:13	Reserved	Reserved	0_0000h	
12	RTTEDM	**Replay Timer Timeout Error Detect Mask [STICKY]:** This bit is sticky through system reset. 0 = Detect Replay Timer Timeout error. 1 = Disable Replay Timer timeout error detection.	0b	RW
11:09	Reserved	Reserved	000b	
08	RNREDM	**REPLAY_NUM Rollover Error Detect Mask[STICKY]:** This bit is sticky through system reset. 0 = Detect REPLAY_NUM rollover 1 = Disable REPLAY_NUM rollover detection.	0b	RW
07	BDEDM	**Bad DLLP Error Detect Mask [STICKY]:** This bit is sticky through system reset. 0 = Detect Bad DLLP error. 1 = Disable Bad DLLP error detection.	0b	RW
06	BTEDM	**Bad TLP Error Detect Mask[STICKY]:** OPTIONAL. This bit is sticky through system reset. 0 = Detect Bad TLP error. 1 = Disable Bad TLP error detection.	0b	RW
05:01	Reserved	Reserved	00000b	
00	REDM	**Receiver Error Detect Mask[STICKY]:** OPTIONAL. This bit is sticky through system reset. 0 = Detect Receiver error. 1 = Disable Receiver Error error detection.	0b	RW

13.4.1.73 Offset 158 - 15Bh: PEAUNITEDMASK – PCI Express Unit Error Detect Mask Register

This register is specific to the IMCH, and controls detection of the PCI Express functional unit error conditions. These bits are sticky through reset.

Table 372. Offset 158 - 15Bh: PEAUNITEDMASK – PCI Express Unit Error Detect Mask Register (Sheet 1 of 2)

Device: 2 **Function:** 0
Offset: 158 - 15Bh **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0000h	
15	UPQODM	**Upstream Posted Queue Overflow Detect Mask[STICKY]:** This bit is sticky through reset. 0 = Enable upstream posted queue overflow detection. 1 = Disable upstream posted queue overflow detection.	0b	RW
14	UNPQODM	**Upstream Non-Posted Queue Overflow Detect Mask[STICKY]:** This bit is sticky through reset. 0 = Enable upstream non-posted queue overflow detection. 1 = Disable upstream non-posted queue overflow detection.	0b	RW

Table 372. Offset 158 - 15Bh: PEAUNITEDMASK – PCI Express Unit Error Detect Mask Register (Sheet 2 of 2)

Device: 2 **Function:** 0
Offset: 158 - 15Bh **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
13	UCQODM	**Upstream Completion Queue Overflow Detect Mask[STICKY]:** This bit is sticky through reset. 0 = Enable completion queue overflow detection. 1 = Disable completion queue overflow detection.	0b	RW
12	LLEPEDM	**LLE Protocol Error Detect Mask[STICKY]:** This bit is sticky through reset. 0 = Enable LLE protocol error detection. 1 = Disable LLE protocol error detection.	0b	RW
11	MDLT	**Mask detection of link transitions from DL_UP to DL_DOWN[STICKY]:** 0 = Enable link down error detection. 1 = Disable link down error detection.	0b	RW
10	DDQPEDM	**Downstream Data Queue Parity Error Detect Mask[STICKY]:** This bit is sticky through reset. 0 = Enable downstream data queue parity error detection. 1 = Disable downstream data queue parity error detection.	0b	RW
09	LUTPEDM	**LUT Parity Error Detect Mask[STICKY]:** This bit is sticky through reset. 0 = Enable LUT parity error detection. 1 = Disable LUT parity error detection.	0b	RW
08	LLRBDPEDM	**LLRB Data Parity Error Detect Mask[STICKY]:** This bit is sticky through reset. 0 = Enable LLRB data parity error detection. 1 = Disable LLRB data parity error detection.	0b	RW
07	LLRBHPEDM	**LLRB Header Parity Error Detect Mask[STICKY]:** This bit is sticky through reset. 0 = Enable LLRB header parity error detection. 1 = Disable LLRB header parity error detection.	0b	RW
06	LLRBCPEDM	**LLRB Control Parity Error Detect Mask[STICKY]:** This bit is sticky through reset. 0 = Enable LLRB control parity error detection. 1 = Disable LLRB control parity error detection.	0b	RW
05	DLLPTEDM	**DLLP Timeout Error Detect Mask[STICKY]:** This bit is sticky through reset. 0 = Enable DLLP Timeout error detection. 1 = Disable DLLP Timeout error detection.	0b	RW
04:03	Reserved	Reserved	0b	
02	MSCLTEDM	**Mask SMB Clock Low Timeout Error Detect Mask [STICKY]:**This bit is sticky through reset. 0 = Enable SMB Clock Low Timeout error detection. 1 = Disable SMB Clock Low Timeout error detection.	0b	RW
01	MUNSEDM	**Mask Unexpected NAK on SMB Error Detect [STICKY]: Mask This** bit is sticky through reset. 0 = Enable Unexpected NAK on SMB error detection. 1 = Disable Unexpected NAK on SMB error detection.	0b	RW
00	MSLBAEDM	**Mask SMB lost Bus Arbitration Error Detect [STICKY]: Mistakes** bit is sticky through reset. 0 = Enable SMB arbitration loss detection. 1 = Disable SMB arbitration loss detection.	0b	RW

13.4.1.74 Offset 160 - 163h: PEAFERR – PCI Express First Error Register

Locks after first error.

Table 373. Offset 160 - 163h: PEAFERR – PCI Express First Error Register (Sheet 1 of 2)

Device: 2	***Function:*** 0
Offset: *160 - 163h*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
31:09	Reserved	Reserved	00_0000h	
08	DFED	**Device Fatal Error Detected [STICKY]:** This bit is for internally detected fatal errors. 0 = No error detected 1 = Error Detected	0b	RWC
07	DNFED	**Device Non-Fatal Error Detected [STICKY]:** This bit is for internally detected non-fatal errors. 0 = No error detected 1 = Error Detected	0b	RWC
06	DCED	**Device Correctable Error Detected [STICKY]:** This bit is for internally detected correctable errors. 0 = No error detected 1 = Error Detected	0b	RWC
05	USFED	**Unit Specific Fatal Error Detected [STICKY]:** This bit is for fatal errors not in the PCI Express specification as logged by the PEAUNITERR register. The PEAMASKERR register only prevents reporting of the unit errors, but does not prevent the logging of errors in this register. 0 = No error detected 1 = Error Detected	0b	RWC
04	USNFED	**Unit Specific Non-Fatal Error Detected [STICKY]:** This bit is for non-fatal errors not in the PCI Express specification as logged by the PEAUNITERR register. The PEAMASKERR register only prevents reporting of the unit errors, but does not prevent the logging of errors in this register. 0 = No error detected 1 = Error Detected	0b	RWC
03	USCED	**Unit Specific Correctable Error Detected [STICKY]:** This bit is for correctable errors not in the PCI Express specification as logged by the PEAUNITERR register. The PEAMASKERR register only prevents reporting of the unit errors, but does not prevent the logging of errors in this register. 0 = No error detected 1 = Error Detected	0b	RWC

Table 373. Offset 160 - 163h: PEAFERR – PCI Express First Error Register (Sheet 2 of 2)

Device: 2	**Function:** 0
Offset: 160 - 163h	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
02	FEMR	**Fatal Error Message received [STICKY]:** This bit is not set for internally detected fatal errors a.k.a. virtual fatal messages. These received fatal error messages can be masked by the SERR enable bit in the Bridge Control Register, if the SERR enable bit is a 0. 0 = No ERR_FATAL message received 1 = An ERR_FATAL message is received.	0b	RWC
01	NFEMR	**Non-fatal Error Message received [STICKY]:** This bit is not set for internally detected non-fatal errors a.k.a. virtual non-fatal messages. These received non-fatal error messages can be masked by the SERR enable bit in the Bridge Control Register, if the SERR enable bit is a 0. 0 = No ERR_NONFATAL message received 1 = An ERR_NONFATAL message is received.	0b	RWC
00	CEMR	**Correctable Error Message received [STICKY]:** This bit is not set for internally detected correctable errors a.k.a. virtual correctable messages. These received correctable error messages can be masked by the SERR enable bit in the Bridge Control Register, if the SERR enable bit is a 0. 0 = No ERR_COR message received 1 = An ERR_COR message is received.	0b	RWC

13.4.1.75 Offset 164 - 167h: PEANERR – PCI Express Next Error Register

Logs errors after FERR register is locks.

Table 374. Offset 164 - 167h: PEANERR – PCI Express Next Error Register

Device: 2	**Function:** 0
Offset: 164 - 167h	**Size:** 32 bit
Default Value: 0000_0000h	

Bits	Name	Description	Reset Value	Access
See Section 13.4.1.74, "Offset 160 - 163h: PEAFERR – PCI Express First Error Register" for bit definitions.				

13.4.1.76 Offset 168 - 16Bh: PEAERRCTL – PCI Express Port A Error Control Register

This register enables the injection of errors on incoming data streams into the core. The lower 16 bits are the corresponding flip parity bits for the cacheline of data. The upper bits in the register are for the use and control of the associated flip parity bits.

Table 375. Offset 168 - 16Bh: PEAERRCTL – PCI Express Port A Error Control Register

Device: 2 **Function:** 0
Offset: 168 - 16Bh **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:20	Reserved	Reserved	00h	
19	SSB	**Stop and Scream bit:** This is a special control for errors going to PCI Express, outgoing from the core. Otherwise outgoing data errors are propagated. Not Supported for PCI Express. 0 = Data errors are propagated. 1 = Data errors are not propagated only reported	0b	RO
18	EDDP	**Enable / Disable data poisoning:** 0 = Disable data poisoning - Errors won't be propagated, only good parity is generated. 1 = Enable data poisoning.	0b	RW
17	FTDP	**Flip the designated parity bits:** 0 = No flip 1 = Flip the designated parity bits (bits 15:00) on all data transfers into the core. If a cacheline is in progress when this register is written, wait until the start of the next cacheline to flip parity bits.	0b	RW
16	FTPBNDT	Flip the parity bits on just the next data transfer: 0 = No flip 1 = Flip the designated parity bits (bits 15:00) in only the next data transfer into the core. If a cacheline is in progress when this register is written, wait until the start of the next cacheline to flip parity bits, in order to ensure all bits flipped are within the same cacheline. Hardware clears this when the injection has been performed. It is possible that the error desired did not occur because the next data transfer was not a complete cacheline, and the error to inject was in a different portion of the CL than was transferred. The hardware still clears the inject error once bit in this case. For completions which do not have the complete address, they are assumed to be 16B aligned addresses and only use implied address bits 03:02 to steer the parity error to the appropriate DW. ***Note:*** Since read completions are a maximum of 32B, half of the injection bits are not utilized.	0b	RWS
15:00	PI	**Parity Inject Bits:** Two bits of parity for each 64 B of data, 16 bits of parity for a cacheline.	0000h	RW

13.5 Device 3, Function 0: PCI Express* Port A1 Standard and Enhanced Registers

Device 3 is the PCI Express Port A1 virtual PCI-to-PCI bridge. Device 2 is PCI Express Port A (in x8 mode) or A0 (in x4 mode). Port A1's associated PCI Express link has a maximum lane width of x4. When Device 2 is configured as a x8 PCI Express link, device 3 is not available. The registers described here include both the standard configuration space and the enhanced configuration space (starting at offset 100h). Except for the registers listed below, all registers for Device 3 are exactly the same as for Device 2.

Note: Hot plug is only supported on Device 2. It is not supported on Device 3.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 376 lists those registers and register bits that are unique to this Device 3 PCI Express port.

Table 376. Device 3, Function 0: PCI Express* Port A1 Standard and Enhanced Registers

Offset		Symbols	Register Name/Function	Default	Access
Start	End				
02h	03h	DID	Device Identification Register	*35B7h*	RO
70h	73h	EXPLNKCAP	PCI Express Link Capabilities Register	*0303_E441h*	RO, RWO
78h	7Bh	EXPSLTCAP	PCI Express Slot Capabilities Register	0000_0000h	RO, RWO

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

13.5.1 Register Details

13.5.1.1 Offset 02 - 03h: DID – Device Identification Register

Table 377. Offset 02 - 03h: DID – Device Identification Register

Device: 3 ***Function:*** 0
Offset: *02 - 03h* ***Size:*** *16 bit*
Default Value: *35B7h*

Bits	Name	Description	Reset Value	Access
15:00	DID	**Device Identification Number:** This is a 16-bit value assigned to the IMCH Device 3, Function 0.	35B7h	RO

13.5.1.2 Offset 70 - 73h: PEA1LNKCAP – PCI Express Link Capabilities Register

This register defines the capabilities of the link.

Table 378. Offset 70 - 73h: PEA1LNKCAP – PCI Express Link Capabilities Register

Device: 3 ***Function:*** 0
Offset: *70 - 73h* ***Size:*** *32 bit*
Default Value: *0303_E441h*

Bits	Name	Description	Reset Value	Access
31:24	PN	**Port Number:** This field indicates the PCI Express port number for the associated PCI Express link.	03h	RO
23:18	Reserved	Reserved	00h	
17:15	Reserved	Reserved	111b	
14:12	Reserved	Reserved	110b	
11:10	Reserved	Reserved	01b	
09:04	MLW	**Maximum Link Width:** This field indicates the maximum width of the PCI Express link. Device 3 reports a value of 000100b indicating a maximum link width of x4. All other encodings are reserved.	00_0100b	RO
03:00	MLS	**Maximum Link Speed:** Hardwired to a value of 1h to indicate a maximum link speed of 2.5 Gbits/s	0001b	RO

13.6 Device 8, Function 0: Extended Configuration Registers

The Extended Configuration Registers comprise Device 8 (D8), Function 0 (F0). Table 379 provides the register address map for this device and function.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 379. Extended Configuration Register Map (D8,F0)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	01h	VID	Vendor Identification Register	8086h	RO
02h	03h	DID	Device Identification Register	35C8h	RO
04h	05h	PCICMD	PCI Command Register	0000h	RO
06h	07h	PCISTS	PCI Status Register	0080h	RO
08h	08h	RID	Revision Identification Register	00h	RO
0Ah	0Ah	SUBC	Sub-Class Code Register	80h	RO
0Bh	0Bh	BCC	Base Class Code Register	08h	RO
0Eh	0Eh	HDR	Header Type Register	00h	RO
2Ch	2Dh	SVID	Subsystem Vendor Identification Register	0000h	RWO
2Eh	2Fh	SID	Subsystem Identification Register	0000h	RWO
B6h	B7h	HPCCTL	hot plug Controller Register	0014h	RO, RW
C8h	CBh	SCRUBLIM	Scrub Limit and Control Register	0000_0000h	RO, RW, RWS
CCh	CFh	SCRBADD	Scrub Address Register	0000_0000h	RO, RW
D0h	D3h	DTCL	DRAM Power Management Control Lower Register	2000_0000h	RO, RW
D4h	D7h	DTCU	DRAM Power Management Control Upper Register	0000_0000h	RO, RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

13.6.1 Register Details

13.6.1.1 Offset 00 - 01h: VID – Vendor Identification Register

The VID Register contains the vendor identification number. This 16-bit register combined with the Device Identification Register uniquely identifies any PCI device. Writes to this register have no effect.

Table 380. Offset 00 - 01h: VID – Vendor Identification Register

Device: 8 ***Function:*** 0
Offset: *00 - 01h* ***Size:*** *16 bit*
Default Value: *8086h*

Bits	Name	Description	Reset Value	Access
15:00	VID	**Vendor Identification:** This register field contains the PCI standard identification for Intel 8086h.	8086h	RO

13.6.1.2 Offset 02 - 03h: DID – Device Identification Register

This 16-bit register combined with the Vendor Identification register uniquely identifies any PCI device. Writes to this register have no effect.

Table 381. Offset 02 - 03h: DID – Device Identification Register

Device: 8 ***Function:*** 0
Offset: *02 - 03h* ***Size:*** *16 bit*
Default Value: 35C8h

Bits	Name	Description	Reset Value	Access
15:00	DID	**Device Identification Number:** This is a 16-bit value assigned to the IMCH Extended Configuration Registers, Device 8.	35C8h	RO

13.6.1.3 Offset 04 - 05h: PCICMD – PCI Command Register

Since IMCH Device 8 does not physically reside on a real PCI bus, this register is not supported.

Table 382. Offset 04 - 05h: PCICMD – PCI Command Register

Device: 8 ***Function:*** 0
Offset: *04 - 05h* ***Size:*** *16 bit*
Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
15:10	Reserved	Reserved	00h	
09	FB2B	**Fast Back-to-Back Enable:** This bit is hardwired to 0.	0b	RO
08	SERRE	**SERR Enable:** This bit is hardwired to 0.	0b	RO
07	ADSTEP	**Address/Data Stepping Enable:** This bit is hardwired to 0.	0b	RO
06	PERRE	**Parity Error Enable:** This bit is hardwired to 0.	0b	RO
05	VGASNOOP	**VGA Palette Snoop Enable:** This bit is hardwired to 0.	0b	RO
04	MWIE	**Memory Write and Invalidate Enable:** This bit is hardwired to 0.	0b	RO
03	SCE	**Special Cycle Enable:** This bit is hardwired to 0.	0b	RO
02	BME	**Bus Master Enable:** This bit is hardwired to 0.	0b	RO
01	MAE	**Memory Access Enable:** This bit is hardwired to 0.	0b	RO
00	IOAE	**I/O Access Enable:** This bit is hardwired to 0.	0b	RO

13.6.1.4 Offset 06 - 07h: PCISTS – PCI Status Register

PCISTS is a 16-bit status register that reports the occurrence of error events on Device 8, Function 0's PCI interface. Since IMCH Device 8 does not physically reside on PCI_A many of the bits are not supported.

Table 383. Offset 06 - 07h: PCISTS – PCI Status Register

Device: 8 ***Function:*** 0
Offset: *06 - 07h* ***Size:*** *16 bit*
Default Value: *0080h*

Bits	Name	Description	Reset Value	Access
15	DPE	**Detected Parity Error:**. Hardwired to 0.	0b	RO
14	SSE	**Signaled System Error:**. Hardwired to 0.	0b	RO
13	RMAS	**Received Master Abort Status:**. Hardwired to 0.	0b	RO
12	RTAS	**Received Target Abort Status:**. Hardwired to 0.	0b	RO
11	STAS	**Signaled Target Abort Status:**. Hardwired to 0.	0b	RO
10:09	DEVT	**DEVSEL Timing:** These bits are hardwired to 00 (fast decode) so that optimum DEVSEL timing for PCI_A is not limited by the IMCH.	00b	RO
08	DPD	**Master Data Parity Error Detected:** Hardwired to 0.	0b	RO
07	FB2B	**Fast Back-to-Back:** This bit is hardwired to '1' (indicating fast back-to-back capability) so that the optimum setting for PCI_A is not limited by the IMCH.	1b	RO
06:00	Reserved	Reserved	0000000b	

13.6.1.5 Offset 08h: RID – Revision Identification Register

Table 384. Offset 08h: RID – Revision Identification Register

Device: 8 ***Function:*** 0
Offset: *08h* ***Size:*** *8 bit*
Default Value: *00h*

Bits	Name	Description	Reset Value	Access
07:00	RID	**Revision Identification Number:** This value indicates the revision identification number for the IMCH Device 8. This number must always be the same as the RID for Device 0, Function 0. 00h = A0 stepping.	00h	RO

13.6.1.6 Offset 0Ah: SUBC – Sub-Class Code Register

Table 385. Offset 0Ah: SUBC – Sub-Class Code Register

Device: 8	***Function:*** 0
Offset: *0Ah*	***Size:*** *8 bit*
Default Value: *80h*	

Bits	Name	Description	Reset Value	Access
07:00	SUBC	**Sub-Class Code:** This value indicates the Sub Class Code into which the IMCH Device 8 falls. 80h = Other system peripheral device.	80h	RO

13.6.1.7 Offset 0Bh: BCC – Base Class Code Register

Table 386. Offset 0Bh: BCC – Base Class Code Register

Device: 8	***Function:*** 0
Offset: *0Bh*	***Size:*** *8 bit*
Default Value: *08h*	

Bits	Name	Description	Reset Value	Access
07:00	BASEC	**Base Class Code:** This value indicates the Base Class Code for the IMCH Device 8. 08h = Other system peripheral device.	08h	RO

13.6.1.8 Offset 0Eh: HDR – Header Type Register

Table 387. Offset 0Eh: HDR – Header Type Register

Device: 8	***Function:*** 0
Offset: *0Eh*	***Size:*** *8 bit*
Default Value: *00h*	

Bits	Name	Description	Reset Value	Access
07:00	HDR	**PCI Header:** Indicates the header type of the IMCH Device 8. 00h = single-function device with standard header layout.	00h	RO

13.6.1.9 Offset 2C - 2Dh: SVID – Subsystem Vendor Identification Register

This value is used to identify the vendor of the subsystem.

Table 388. Offset 2C - 2Dh: SVID – Subsystem Vendor Identification Register

Device: 8	***Function:*** 0
Offset: *2C - 2Dh*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:00	SUBVID	**Subsystem Vendor ID:** This field must be programmed during boot-up to indicate the vendor of the system board.	0000h	RWO

13.6.1.10 Offset 2E - 2Fh: SID – Subsystem Identification Register

This value is used to identify a particular subsystem.

Table 389. Offset 2E - 2Fh: SID – Subsystem Identification Register

Device: 8	***Function:*** 0
Offset: *2E - 2Fh*	***Size:*** *16 bit*
Default Value: *0000h*	

Bits	Name	Description	Reset Value	Access
15:00	SUBID	**Subsystem ID:** This field must be programmed during BIOS initialization.	0000h	RWO

A 16-bit PILOT bus can be sampled by the SNAPTRIG register, steered to CHAP for counting purposes, and the lower byte driven out to device pins. There is a hierarchy of multiplexers within IMCH to achieve this functionality. Each unit that has a pilot Mix has its own register (PILOT-unit) to select on an upper and lower byte basis what is put onto its unit output pilot 2-byte bus. Within the T unit the PILOTMUX register selects between the different unit pilot busses. It is the output of this global mux which can be observed by internal logic and presented to the pins. Different bits of this register is used to configure the output path of the pilot signals to the debug pins, to select between chap counter outputs, to select clock from HSI and also to select flopped enable and flopped data for pilot signals.

13.6.1.11 Offset B6 - B7h: HPCCTL – Hot Plug Controller Control Register

The bits in this register define how the hot plug controller behavior can change.

Table 390. Offset B6 - B7h: HPCCTL – Hot Plug Controller Control Register

Device: 8	***Function:*** 0
Offset: *B6 - B7h*	***Size:*** *16 bit*
Default Value: *0014h*	

Bits	Name	Description	Reset Value	Access
15:6	Reserved	Reserved	0b	
05:04	IOESM	**I/O Expander Support Mode:** 00 Reserved 01 Single Byte mode Low (Device address inputs 2:0 forced to 0) 10 Single Byte mode High (Device address inputs 2:1 forced to 0, device address input 0 forced to 1) 11 Reserved	01b	RW
03:00	SMBUAD	**SMB Upper Address Bits:** These bits are equivalent to device address inputs six down to three. Device address inputs two and one are expected to be zero. Device address input 0 must conform to the description for the I/O expander support modes. Defaults to address bits of Philips* 9554 I/O Expander.	0100b	RW

13.6.1.12 Offset C8 - CBh: SCRUBLIM – Scrub Limit and Other Control Information Register

This register is used to load the scrub engine with a particular limit address and other control information, such as the initialization data pattern.

Table 391. Offset C8 - CBh: SCRUBLIM – Scrub Limit and Other Control Information Register

Device: 8	***Function:*** 0
Offset: *C8 - CBh*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
31	SCRBVLD	**Scrublim Valid:** When this bit is written to '1' by software, the contents of this register take affect on the next scrub address update. The hardware clears this bit back to '0' as it loads these values, therefore software should never expect to read '1' on this bit.	0b	RWS
30:29	Reserved	Reserved	00b	
28	MSKPS	**Mask periodic scrubbing:** 0 = Writes for the periodic scrubs are performed 1 = Writes for the periodic scrubs are not performed	0b	RW
27	MSKDS	**Mask demand scrubbing:** 0 = Writes for the demand scrubs are performed 1 = Writes for the demand scrubs are not performed	0b	RW
26:15	Reserved	Reserved	000h	
14:00	LMTAD	**Limit Address:** Defines address bits [34:20] to limit the top of the address range used for scrubs.	0000h	RW

13.6.1.13 Offset CC - CFh: SCRBADD – Scrub Address Register

This register is used to load the scrub engine with a particular starting address. The scrub is performed between this address and the address in the scrub limit register.

Table 392. Offset CC - CFh: SCRBADD – Scrub Address Register

Device: 8	***Function:*** 0
Offset: *CC - CFh*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
31	SCRBADVB	**Scrbadd valid bit:** 0 = The hardware clears this bit as it loads the scrub address from bits 27:00, therefore software should never expect to read '1' on this bit 1 = The value of the address in bits 27:00 are loaded into the scrub unit on the next scrub address update. **NOTE:** The address is loaded at the counter rollover time, which isn't externally visible. Therefore it is possible that after this register is written, there is one more scrub using the internal counter address, instead of the address placed in this register. There is never more than one 'extra' scrub before the value of this register is used.	0b	RW
30:29	Reserved	Reserved	00b	
28:00	SSAD	**Starting scrub address:** This corresponds to address bits 34:06 of the scrub address, which is always 64 B line based.	000_0000h	RW

13.6.1.14 Offset D0 - D3h: DTCL – DRAM Throttling Control Lower Register

DTCL and DTCU make up a conceptual 64-bit register. This throttling mechanism is used for activates, reads, and writes. The memory subsystem uses this register to apply on a per DIMM slot basis. Once throttling is invoked, no transactions are issued to the entire memory subsystem.

Table 393. Offset D0 - D3h: DTCL – DRAM Throttling Control Lower Register

Device: 8	***Function:*** 0
Offset: *D0 - D3h*	***Size:*** *32 bit*
Default Value: *2000_0000h*	

Bits	Name	Description	Reset Value	Access
31:30	Reserved	Reserved	00b	
29:28	TW	**Transaction Weighting:** These two bits select the weighting between activate and read/write commands. A read or write increments the throttle counter by two. The activate command increments the counter by the amount specified by this 2-bit field. In a given cycle, a read or write can occur along with an activate to a different DIMM. Two different DIMM slot counters increment for this cycle. Activate: read/write ratio. 00 2:2 (Increment by 2 for an activate, increment by 2 for a read or write) 01 3:2 (Increment by 3 for an activate, increment by 2 for a read or write) 10 4:2 (Increment by 4 for an activate, increment by 2 for a read or write) 11 5:2 (Increment by 5 for an activate, increment by 2 for a read or write)	10b	RW
27:22	TT	**Throttle Time:** This value provides a multiplier between 0 and 63, which specifies how long throttling remains in effect as a number of Global DRAM Sampling Windows. For example, if GDSW is programmed to 1000_0000b and TT is set to 01_0000b, then throttling is performed for ~8 seconds once invoked (128 * 4ms * 16).	00h	RW
21:15	TMW	**Throttle Monitoring Window:** The value in this register is shifted left by 4 to specify a window of 0-2047 host clocks with a 16 clock granularity. While the throttling mechanism is invoked, DRAM activate, reads, and writes are monitored during this window. If the weighted activity count during the window reaches the Throttle Activity Maximum for any DIMM slot, then requests are blocked for the remainder of the window for all DIMM slots.	00h	RW
14:03	TAM	**Throttle Activity Maximum:** This value defines the maximum weighted activity count, between 0–4095, which is permitted to occur on a DIMM slot within one TMW.	000h	RW
02:01	RM	**Throttle Mode:** 00 Throttling via counters and hardware throttle_on signal mechanisms disabled. 01 reserved 10 Counter mechanism controlled through GDSW and GAT is enabled. When the threshold set in GDSW and GAT is reached, throttling start/stop cycles occur based on the setting in TT, TMW and TAM. 11 reserved	00b	RW
00	ST	**Start Throttle:** Software writes to this bit to start and stop write throttling. 0 = Write throttling stops and the counters associated with TMW and TAM are reset. 1 = Write throttling begins based on the settings in TMW and TAM; and remains in effect until this bit is reset to '0'.	0b	RW

13.6.1.15 Offset D4 - D7h: DTCU – DRAM Throttling Control Upper Register

DTCL and DTCU make up a conceptual 64-bit register.

Table 394. Offset D4 - D7h: DTCU – DRAM Throttling Control Upper Register

Device: 8	***Function:*** 0
Offset: *D4 - D7h*	***Size:*** *32 bit*
Default Value: *0000_0000h*	

Bits	Name	Description	Reset Value	Access
31:30	TLOCK	**Throttle Lock:** These bits secure the DRAM throttling control registers. The bits default to '0'. Once a '1' is written to either bit, the configuration register bits in DTC become read-only, and no later writes can change the register until reset. An exception to this statement for ST is noted below. ***Note:*** This register is not sticky. 00 Not locked. All of the bits in DTC can be written. 01 START Mode bits not locked. All bits in DTC(U and L) except for the ST (DTCL bit 0) is locked and cannot be written to. Note that bits 31:30 are also locked, just as when the throttle lock bits are "10". 10 All bits locked. DTC is fully locked and cannot be changed, including bits 31:30 of this register. 11 reserved	00b	RW
29	TME	**Throttle Test Mode Enable:** This bit is used to shorten test time. 0 = Normal operation 1 = Global DRAM Sampling Window, and the Global Activate Threshold are scaled down by ~1000. GDSW becomes a specification of microseconds rather than milliseconds and GAT is multiplied by 32 rather than 32k.	0b	RW
28:21	Reserved	Reserved	00h	
20:13	GDSW	**Global DRAM Sampling Window:** This 8b value is multiplied by four to define the length of time in milliseconds (0–1020). If the weighted activity count during this window exceeds the Global Activity Threshold defined below, then the throttling mechanism is invoked to limit DRAM requests to a lower bandwidth checked over smaller time windows across all DIMM slots.	00h	RW
12:00	GAT	**Global Activity Threshold:** This 13b value is multiplied by 2^{15} to arrive at the weighted activity count that must occur on a DIMM slot within the Global DRAM Sampling Window in order to cause the throttling mechanism to be invoked.	000h	RW

13.7 Memory Mapped I/O for DDR2 Registers

This section describes the memory-mapped registers for the Memory Controller. The SMRBASE register, described in Section 13.1, "Device 0, Function 0: IMCH Registers"provides the base address for these registers. The offsets listed for the following registers are relative to this base address.

The value for BAR for all registers in this section is BAR14h.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure or undetermined behavior.

Note: Reserved bits are Read Only.

Table 395. Memory Mapped I/O for DDR2 Register Summary

Offset		Symbol	Register Name/Function	Sticky	Default	Access
Start	End					
100h	103h	DCALCSR	DCAL Control and Status Register	No	*0000_0000h*	RW
104h	107h	DCALADDR	DCAL Address Register	No	*0000_0000h*	RW
108h	10Bh	DCALDATA0	DCAL Data Register DW0	No	*0000_0000h*	RW
10Ch	10Fh	DCALDATA1	DCAL Data Register DW1	No	*0000_0000h*	RW
110h	113h	DCALDATA2	DCAL Data Register DW2	No	*0000_0000h*	RW
114h	117h	DCALDATA3	DCAL Data Register DW3	No	*0000_0000h*	RW
118h	11Bh	DCALDATA4	DCAL Data Register DW4	No	*0000_0000h*	RW
11Ch	11Fh	DCALDATA5	DCAL Data Register DW5	No	*0000_0000h*	RW
120h	123h	DCALDATA6	DCAL Data Register DW6	No	*0000_0000h*	RW
124h	127h	DCALDATA7	DCAL Data Register DW7	No	*0000_0000h*	RW
128h	12Bh	DCALDATA8	DCAL Data Register DW8	No	*0000_0000h*	RW
12Ch	12Fh	DCALDATA9	DCAL Data Register DW9	No	*0000_0000h*	RW
130h	133h	DCALDATA10	DCAL Data Register DW10	No	*0000_0000h*	RW
134h	137h	DCALDATA11	DCAL Data Register DW11	No	*0000_0000h*	RW
138h	13Bh	DCALDATA12	DCAL Data Register DW12	No	*0000_0000h*	RW
13Ch	13Fh	DCALDATA13	DCAL Data Register DW13	No	*0000_0000h*	RW
140h	143h	DCALDATA14	DCAL Data Register DW14	No	*0000_0000h*	RW
144h	147h	DCALDATA15	DCAL Data Register DW15	No	*0000_0000h*	RW
148h	14Bh	DCALDATA16	DCAL Data Register DW16	No	*0000_0000h*	RW
14Ch	14Fh	DCALDATA17	DCAL Data Register DW17	No	*0000_0000h*	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

13.7.1 Register Details

13.7.1.1 Offset 00 - 01h: NOTESPAD – Note (Sticky) Pad for BIOS Support Register

This dedicated 16-bit register is provided for BIOS.

Table 396. Offset 00 - 01h: NOTESPAD – Note (Sticky) Pad for BIOS Support Register

Offset: 00 - 01h *Default Value: 0000h*		*Size: 16 bit*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
15:00	BSR	**BIOS sticky register [STICKY]:** This register is used by BIOS. It is sticky through reset.	0000h	RW

13.7.1.2 Offset 02 - 03h: NOTEPAD – Note Pad for BIOS Support Register

This dedicated 16 bit register is provided for BIOS.

Table 397. Offset 02 - 03h: NOTEPAD – Note Pad for BIOS Support Register

Offset: 02 - 03h *Default Value: 0000h*		*Size: 16 bit*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
15:00	BNSR	**BIOS non-sticky register:** This register is used by BIOS. It is not sticky through reset.	0000h	RW

13.7.1.3 Offset 100 - 103h: DCALCSR – DCAL Control and Status Register

This 32-bit register controls and shows status for the DCAL operation.

Figure 75. DCAL Control and Status Register

Bit	DCALCSR(31:0) Definition										
31	W = Start, R = In Progress										
30	Pass/Fail Indicators : 000 = Pass, x01 = Channel Access Denied, x10 = Unpopulated Row Selected, x11 = Unsupported Opcode, 1xx = Operation Completed with a Failure										
29											
28											
27	Quiesce Request : 00 = Normal operation, 01,10,11 = Reserved										
26											
25	Channel Select : 00 = None, 01 = Memory channel, 10, 11 = Reserved										
24											
23	Operation mode : 0 = One pass, 1 = All passes										
22	Row Select (Chip Select)										
21											
20											
19	Refresh (1 = Enabled, 0 = Disabled)										
18	Fixed Data Pattern Selection										
17											
16											
15	Disable Read/Write Pointer Reset (for debug only)										
14	-	-	-	-	-	DLL Slave Length	Single Pass DLL Slave Length and Delay	-	-	-	-
13	-	-	-	Cycle Wait Before Collecting OCD Sample	-			-	FERR enable		-
12	-	-	-		-			Pass2	Log Select	-	-
11	-	-	-		-	Single Pass DLL Delay		Cycle Wait to Allow DLLs to Lock	Halt on Error	-	-
10	-	-	-		-				LFSR Seed	Read Pointer	-
9	-	-	-		Receive Enable Delay used in Single Pass				Data Select		-
8	-	-	-								-
7	-	-	-			Cycle Count for Left Edge of Data Eye	-		Test Type		-
6	-	-	-				Receive Enable Delay			RTRY	-
5	-	-	-						Write, Read Compare	DED	-
4	-	-	-			Logical Addr				SEC	-
3	Opcode(3:0)										
2			Pre-	MRS /	Receive			DLL		Error	No
1	NOP	Refresh	Charge	EMRS	Enable	DQS Cal	I/O Lpbk	BIST	Memory Test	Monitor	Command
0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1110	1111

Table 398. Offset 100 - 103h: DCALCSR – DCAL Control and Status Register (Sheet 1 of 5)

Offset: *100 - 103h* ***Size:*** *32 bit*

Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31	SRTINP	**Start / In Progress:** This bit is cleared by hardware when the DCAL operation has completed. 0 = No DCAL operation in progress 1 = Set by software to initiate a DCAL operation. Set bit indicates operation in progress.	0b	RW

Table 398. Offset 100 - 103h: DCALCSR – DCAL Control and Status Register (Sheet 2 of 5)

Offset: 100 - 103h **Size:** 32 bit

Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
30:28	PFIND	**Pass/Fail indicators:** 000 Test passed 001 Access to the memory channel was denied. This error occurs when the memory channel is "owned" by the normal operation controller and channel borrowing was not selected. 010 The selected row is not populated. 011 The selected opcode is not supported. Opcodes 0x0 through 0x8 are defined and opcodes 0x9 through 0xF are reserved (not supported). 100 The operation failed with no additional information. 101 The operation had a memory channel failure. 110 Reserved 111 Reserved Special encodings for Error Monitor DED Retry Mode. 000 The operation is waiting for a DED error or the retry data. X01 The operation completed and the retry data had a data error. X10 Reserved. X11 Reserved. 100 The operation completed and the retry data had no data errors.	000b	RW
27:26	QREQ	**Quiesce Request:** 00 Normal mode of operation, used when the channel arbiter is in the IDLE state. 10 Reserved 11 Reserved.	00b	RW
25:24	CHANSEL	**Channel Selection:** 00 No channel selected. This causes a "001" failure (listed above). 01 Memory channel selected. 10 Reserved 11 Reserved	00b	RW
23	OPMODE	**Operation mode:** This field is only applicable to Receive Enable Calibration, DQS Calibration, AC I/O Loopback and Memory Test operations: 0 = Single pass 1 = All passes	0b	RW
22:20	ROWSEL	**Row Select:** 000: Logical row 0(CS0nn before the DRM remapping) 001: Reserved 010: Logical row 1(CS2nn before the DRM remapping) 011: Reserved 100: Logical row 2(CS4nn before the DRM remapping) 101: Reserved 110: Logical row 3(CS6nn before the DRM remapping) 111: Reserved	000b	RW
19	RFEN	**Refresh Enable:** 0 = Refresh controller disabled 1 = Refresh controller enabled	0b	RW

Table 398. Offset 100 - 103h: DCALCSR – DCAL Control and Status Register (Sheet 3 of 5)

Offset: *100 - 103h* ***Size:*** *32 bit*

Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
18:16	FXDPS	**Fixed (Nibble) Data Pattern Selection**: This is only applicable for the Memory Test, DQS Calibration and AC I/O Loopback opcodes: Encoding — Write data / Expected Read Data (replicated across all nibbles) 000 — F -> 0 -> F -> 0 001 — 0 -> F -> 0 -> F 010 — A -> 5 -> A -> 5 011 — 5 -> A -> 5 -> A 100 — C -> 3 -> C -> 3 101 — 3 -> C -> 3 -> C 110 — 9 -> 6 -> 9 -> 6 111 — 6 -> 9 -> 6 -> 9	000b	RW
15	Reserved	Reserved.	0b	
14:04	Opcode Dependent Qualifiers	**Opcode Dependent Qualifiers:** The definition of these bits is opcode dependent: NOP, Refresh, Pre-Charge, MRS and No Command EMRS: 14 Reserved, not used. 13:04 OCD Sample Wait: the number of (uwclk) cycles to wait before collecting the OCD Drive(0)/Drive(1) samples. This field is only used if an EMRS operation (DCALADDR, BA = 001b) is selected and either the OCD Drive(0) or Drive(1) command is selected (DCALADDR, MA(9:7) = 010b or 001b). Receive Enable Calibration: 14:10 Reserved, not used. 09:04 Single Pass Receive Enable Delay: these bits contain a 6-bit receive enable delay that is only used if a single-pass operation is selected. The all-pass operation automatically cycles through all 64 possible encodings of the receive enable delay. DQS Calibration: 14:12 DLL slave length, used to set the coarse DLL delay adjustment, used in both single pass and all-pass modes. 11:08 DLL slave mix, used to set the fine DLL delay adjustment. This field is only used in the single pass mode. The all-pass mode automatically cycles through all 16 possible encodings of the DLL slave mix. 07:05 Cycle count for the filter that detects the start of the left edge of the DQ data eye. When set to X, it takes X+1 consecutive passes (no fails) before the left edge of the data eye is detected. This is needed only on the left edge because the data eye scan proceeds from left to right (increasing DLL delays). 04 Logical address enable bit. When set to a 1, the DQS calibration operation uses a logical address instead of a physical address.	000h	RW

Table 398. Offset 100 - 103h: DCALCSR – DCAL Control and Status Register (Sheet 4 of 5)

Offset: 100 - 103h **Size:** 32 bit

Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
14:04 (continued)	Opcode Dependent Qualifiers	Memory Test: 14 Reserved, not used. 13 FERR/NERR enable: When set (1), bit 15 of FERR (or NERR in the case that FERR is already non-zero) is set at the conclusion of the Memory Test. When cleared to 0, then FERR/NERR is not affected. The memory test operation can take about 15 seconds to complete for the lowest frequency and the maximum memory size (16 Gbyte). 12 Error Log Select: This bit is used when the upper half of DCALDATA is used as the write/read-compare data and the lower half of DCALDATA is used as an error log. When this bit is set to a 1, the lower half of DCALDATA is used as an error address log, in which a non-zero Dword entry in DCALDATA Dwords 0-7 and 16 indicate an address that encountered a read-compare error. This first error is logged in Dword 0, then Dword 1, ..., then Dword 7 and then Dword 16. When this bit is cleared to a 0, the lower half of DCALDATA is used as a data bit line failure accumulator. If any bit gets set in Dwords 0-7 and 16, it indicates a failure occurred in the associated bit lane on one or more of the read-compares. For fixed data patterns, the upper half of DCALDATA is used as the error address log and the lower half of DCALDATA is used as the bit lane failure accumulator. 11 Halt on error: When set to a 1, the operation halts after a read-compare error is detected. When cleared to a 0, the operation does not halt due to a detected error. 10 LFSR seed: This bit is used only when the Data Select field selects the LFSR to generate the write/read-compare data. When set to a 1, DCALDATA Dword 17 is used as the LFSR seed; otherwise a seed of all zeroes is used. 09:08 Data Select (2 bits): 00 = Fixed data pattern selected by DCALCSR bits 18:16. 01 = DCALDATA (full cache line) is used as the data pattern. 10 = LFSR Data: DCALDATA Dword 17 contains the LFSR data generator, Dwords 8-15 get updated from the swizzled data value in the next higher Dword. 11 = Circular Swizzle Data: DCALDATA Dwords 8-15, 17 are used as the data pattern, with a circular sizzle shift; the data pattern repeats after 288 data phases.	000h	RW

Table 398. Offset 100 - 103h: DCALCSR – DCAL Control and Status Register (Sheet 5 of 5)

Offset: 100 - 103h **Size:** 32 bit

Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
14:04 (continued)	Opcode Dependent Qualifiers	07:06 Test Type: 00 = Single/Logical Address: The operation is performed to a single address that is contained in DCALADDR, which is interpreted as a logical address (same decoding as a B Unit initiated transaction). 01 = Single/Physical Address: The operation is performed to a single address that is contained in DCALCSR/DCALADDR, which is interpreted as a physical address (Row select in DCALCSR and row/column/bank addresses in DCALADDR). 10 = Full Row: The selected row is specified in DCALCSR. The starting address is defined by DRBX-1 and the ending address is defined by (DRBX)–1. If the selected row is 0, then the starting address is logical address 0x0. 11 = All of Memory: Starting address is logical address 0x0 and the ending address is defined by (DRB7)-1. 05:04 Write/Read/Compare: 00 = Read only; no writes or data comparisons are performed 01 = Write only; no reads or data comparisons are performed. 10 = Read with compare; no writes are performed. 11 = Write then read with compare. Error Monitor: 14 Reserved, not used. 13 FERR/NERR enable: When set (1), bit 15 of FERR (or NERR in the case that FERR is already non-zero) is set when the error monitor detects an error. 12:11 Reserved, not used. 10:08 Read Pointer: This field is used only when none of the 3 error types are selected to monitor. If bits 6:4 are all zero, this read pointer is sent to the DDR2 in order to read the selected buffer in the read data FIFO. Note that read pointers 0 through 4 access the read data FIFO, read pointers 5 through 7 access the write pointers. Read pointer 5 also accesses the DLL BIST clocks, the OCD calibration samples and any other information routed to the available MUX inputs on read pointer position 5. 07 Reserved. 06 Reserved. 05 Uncorrectable error (DED) monitor enable: When set (1), the error monitor monitors the DED error signal from DDP/DDISP. 04 Single-bit error (SEC) monitor enable: When set (1), the error monitor monitors the SEC error signal from DDP/DDISP.	000h	RW
03:00	Opcode	These bits select the operation: 0000 NOP: a single NOP is issued to the select row. 0001 Refresh: a single auto-refresh is issued to the selected row. 0010 Pre-charge: a single pre-charge (all banks) is issued to the select row. 0011 MRS/EMRS: a single (E)MRS command is issued to the selected row. 0100 Receive enable calibration operation. 0101 DQS calibration operation 1000 Memory test/initialization operation. 1110 Error Monitor 1111 No command Others Reserved, not supported.	0h	RW

13.7.1.4 Offset 104 - 107h: DCALADDR – DCAL Address Register

This 32-bit register supplies address and other information for the DCAL operation, depending on the opcode selected in the DCAL control and status register.

Figure 76. Offset 104 - 107h: DCALADDR – DCAL Address Register

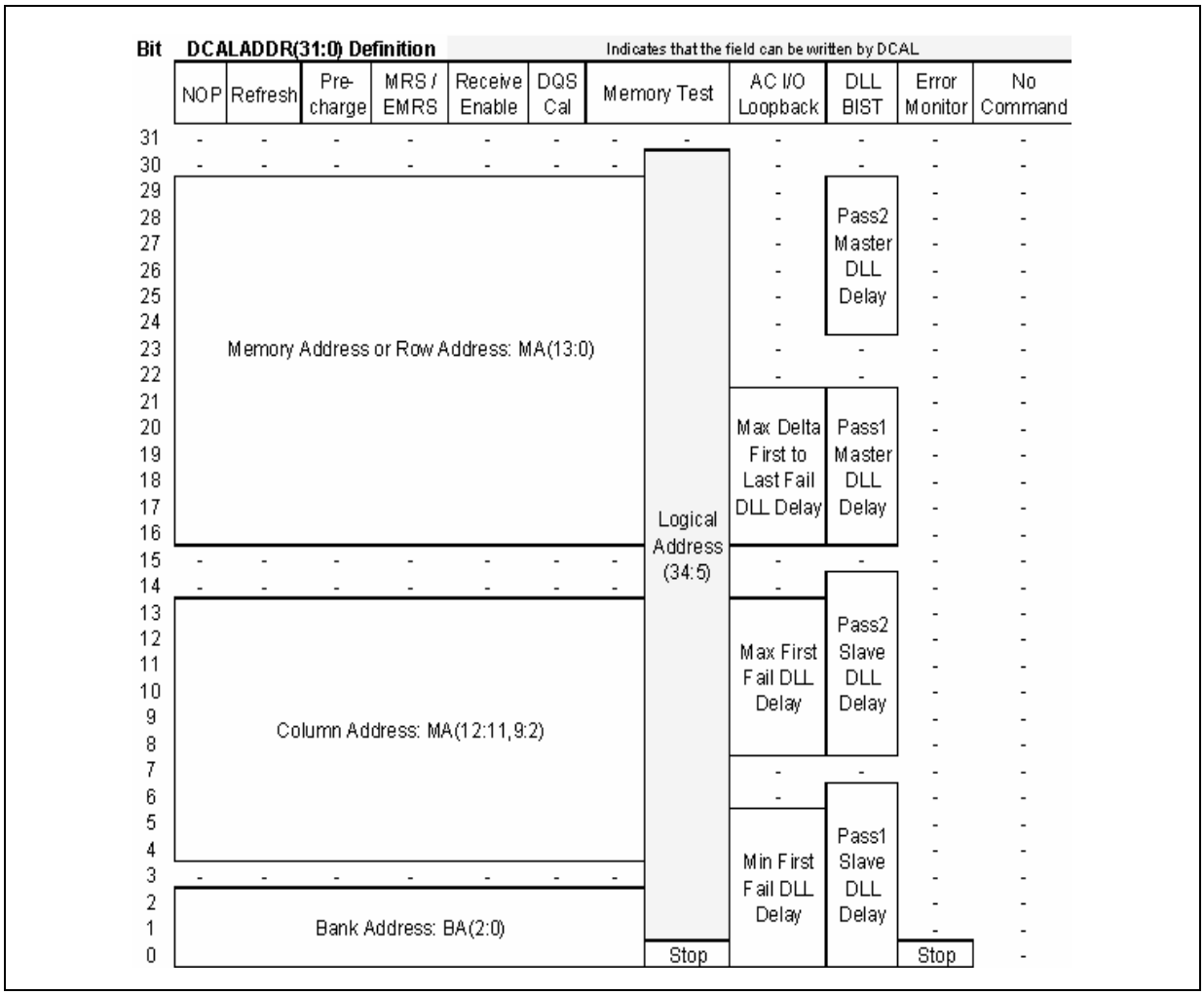


Table 399. Offset 104 - 107h: DCALADDR – DCAL Address Register

Offset: 104 -107h **Size:** 32 bits

Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:00	DCAL Opcode dependent address definition	The text below breakdowns this register based on the opcodes from DCALCSR. Opcodes: NOP, Refresh, Pre-charge, MRS/EMRS, Receive enable cal, DQS cal (when logical addressing is not enabled; DCALCSR(4)=0), Memory Test (when the test type is single/ physical address; DCALCSR(7:6)=01): 31 Reserved, not used. 30:16 MA(14:0): This is the 15-bit address to be driven on the DDR_MA(14:0) IMCH pins for the activate (row) command for the Receive enable calibration, DQS calibration and Memory Test read/writes and for the single commands that are issued (NOP, Refresh, Pre-charge all banks, MRS/EMRS. 15:14 Reserved, not used. 13:4 MA(12:11,9:2): This field is used to construct the 15-bit address to be driven on the DDR_MA(14:0) IMCH pins for the read/write (column) command for Receive enable calibration, DQS calibration and Memory Test read/writes. 3 Reserved, not used. 2:0 BA(2:0): This is the 3-bit bank address to be driven on the DDR_BA(2:0) IMCH pins for all commands. Opcodes: DQS cal (when logical addressing is enabled; DCALCSR(4)=1), Memory Test (when the test type is not single/physical address; DCALCSR(7:6)=00, 10 or 11) 31 Reserved, not used. 30:1 Logical Address: This value corresponds to bits 34:5 of the logic address (identical to the address that would be received from the B Unit after the PCI-hole address remapping). This field is used by DQS calibration as a read-only value for the address to perform the write/reads. This field is also used by the memory test operation to hold the current logical address, which increments after each write/ read until the end of the test is reached (just past the end of the select row or equal to DRB7 in the case of the all memory operation).	0000_0000h	RW
31:00 (Continued)	DCAL Opcode dependent address definition	0 Stop: This bit is used by the Memory Test operation (not used by DQS calibration) to immediately halt the “full row” or “all memory” test. In the case of a write/read-compare test, this bit has to be written twice, the first time to halt the writes and the second time to halt the reads. This bit is only intended to be used in Pre-Si validation and Post-Si debug. For normal usage, this bit must be 0. **Opcode: Error Monitor** 31:1 Reserved, not used. 0 Stop: When this bit is set, the error monitor stops and DCALCSR(31) clears. Note that the Error Monitor can run indefinitely (due to the absence of errors or when all-pass mode is selected) and DCALCSR is not writable when an operation is in progress. **Opcode: No command** **31:0** Reserved, not used.	00	RW

13.7.1.5 Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0

This 32-bit register is one of 18 data registers used to support the various DCAL engine opcodes. The definition of these bits changes with the DCAL Address Register changes depending on the operation.

Table 400. Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0 (Sheet 1 of 7)

Offset: *108 - 10Bh* ***Size:*** *32 bits*

Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:00	DCAL Data	**DCAL Data:** Definition opcode dependent. **OPCODE: NOP, Refresh, MRS, pre-charge all banks and No Command** Reserved, not used for all 18 Dwords. **OPCODE: EMRS (OCD Adjust/Drive Commands)** Adjust: A 4-bit value stored as $0000D_{T3}D_{T2}D_{T1}D_{T0}$ within the byte, which is the 4-bit data burst for the OCD adjust command. Notice the bit reversal; bit 0 is in the left-hand column. All other encodings are reserved.	0000_0000h	RW

4-bit burst codes inputs to all DQs				Operation	
D_{T0}	D_{T1}	D_{T2}	D_{T3}	Pull-up driver strength	Pull-down driver strength
0	0	0	0	NOP (No operation)	NOP (No operation)
0	0	0	1	Increase by 1 step	NOP
0	0	1	0	Decrease by 1 step	NOP
0	1	0	0	NOP	Increase by 1 step
1	0	0	0	NOP	Decrease by 1 step
0	1	0	1	Increase by 1 step	Increase by 1 step
0	1	1	0	Decrease by 1 step	Increase by 1 step
1	0	0	1	Increase by 1 step	Decrease by 1 step

Other Combinations Reserved

Drive: An 8-bit value stored as $D_{1S1}D_{1S2}D_{1S3}D_{1S4}D_{0S1}D_{0S2}D_{0S3}D_{0S4}$ within the byte. The D_{1Sx} values represent the 4 samples collected during the EMRS OCD Drive(1) operation and the D_{0Sx} values represent the 4 samples collected during the EMRS OCD Drive(0) operation. After running the Drive(1) and Drive(0) commands, this 8-bit value must be 0xFF to guarantee that the drive strengths are sufficient. If the Drive(1) samples do not equal 0xF, then the pull-up driver strength needs to be increased by (at least) 1 step with the OCD adjust command. If the Drive(0) samples do not equal 0xF, then the pull-down driver strength needs to be increased by (at least) 1 step with the OCD adjust command.

Table 400. Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0 (Sheet 2 of 7)

Offset: 108 - 10Bh **Size:** 32 bits

Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:00 (Cont'd)	DCAL Data	**OPCODE: Receive enable calibration.** Write pointers (wrptr): A 5-bit value stored as 000xxxxx within the byte. The write pointers should be a one-hot encoded value. The legal values are 0x01, 0x02, 0x04, 0x08 and 0x10. Sampled High/Low Vectors: A 64-bit value stored in 2 consecutive dwords that contain 1-bit per pass of the receive enable operation. Bit 0 represents the results from pass 0, bit 1 represents the results from pass 1, and so forth. By comparing the sampled high vector to the sampled low vector, the rising/falling edges of DQS can be located. In general, the pattern in each vector should repeat every 8 bits, except where the preamble is located. Legal values: bit X of one (or both) of the sampled high and low vectors must be equal to zero (both should never be equal to one). In other words, a bit-wise AND of the 2 vectors should results in all zeroes. **DW(s) Bit(s) Description** 17 31:24 Expected write pointer when DQS is sampled high. This is when the 1 cycle wide rcvcal pulse ANDed with the DQS signal produces two small pulses that cause the write pointers to advance twice. The initial value of the write pointer is 0x01, so this value must be programmed to 0x04. 17 23:16 Expected write pointer when DQS is sampled low. This is when the 1 cycle wide rcvcal pulse ANDed with the DQS signal produces a single pulse that cause the write pointers to advance once. The initial value of the write pointer is 0x01, so this value must be programmed to 0x02. 17 15:08 Reserved 17 07:00 The OR'd wrptr: All of the write pointers are logically OR'ed to produce this value, which is then compared to the expected sampled high/low write pointers to produce the sampled high/low vectors stored in DCALDATA(03:00)(31:00).	0000_0000h	RW

Table 400. Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0 (Sheet 3 of 7)

Offset: *108 - 10Bh* ***Size:*** *32 bits*

Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:00 (Cont'd)	DCAL Data		0000_0000h	RW

DW(s)	31:24 (Byte 3)	23:16 (Byte 2)	15:8 (Byte 1)	7:0 (Byte 0)
16	Reserved	Reserved	Reserved	wrptr DQS8
15	Reserved	Reserved	Reserved	Reserved
14	Reserved	Reserved	Reserved	Reserved
13	Reserved	Reserved	Reserved	Reserved
12	Reserved	Reserved	Reserved	Reserved
11	wrptr DQS16	wrptrDQS7	wrptr DQS15	wrptr DQS6
10	wrptr DQS14	wrptr DQS5	wrptr DQS13	wrptr DQS4
09	wrptr DQS12	wrptr DQS3	wrptr DQS11	wrptr DQS2
08	wrptr DQS10	wrptrDQS1	wrptrDQS9	wrptrDQS0
07	Reserved			
06	Reserved			
05	Reserved			
04	Reserved			
03	Cumulative "Sampled High" vector for passes(63:32)			
02	Cumulative "Sampled High" vector for passes(31:00)			
01	Cumulative "Sampled Low" vector for passes(63:32)			
00	Cumulative "Sampled Low" vector for passes(31:00)			

Table 400. Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0 (Sheet 4 of 7)

Offset: *108 - 10Bh* ***Size:*** *32 bits*

Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:00 (Cont'd)	DCAL Data	**OPCODE: DQS calibration.** Pass: A 16-bit vector of Pass=1/Fail=0 results; 1-bit per pass of the DQS calibration operation. Bit 0 represents the results from pass 0 (DLL slave mix = 0), Bit 1 represents the results from pass 1 (DLL slave mix = 1), and so forth. Fields left blank are reserved.	0000_0000h	RW

DW(s)	31:16 (Word 1)	15:0 (Word 0)
17	Reserved	Pass DQS17
16	Reserved	Pass DQS8
15	Reserved	Reserved
14	Reserved	Reserved
13	Reserved	Reserved
12	Reserved	Reserved
11	Pass DQS16	Pass DQS14
10	Pass DQS12	Pass DQS10
09	Pass DQS15	Pass DQS13
08	Pass DQS11	Pass DQS9
07	Reserved	Reserved
06	Reserved	Reserved
05	Reserved	Reserved
04	Reserved	Reserved
03	Pass DQS7	Pass DQS5
02	Pass DQS3	Pass DQS1
01	Pass DQS6	Pass DQS4
00	Pass DQS2	Pass DQS0

Table 400. Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0 (Sheet 5 of 7)

Offset: *108 - 10Bh* ***Size:*** *32 bits*

Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:00 (Cont'd)	DCAL Data	**OPCODE: Memory Test (Data Type is DCALDATA).** All of DCALDATA is used as the cache line of write data and to capture a single cache line of read data. Fields left blank are reserved. Top 16 bits reserved	0000_0000h	RW

DW(s)	31:24 (Byte 3)	23:16 (Byte 2)	15:8 (Byte 1)	7:0 (Byte 0)
17	Reserved	Reserved	RW CB burst 3	RW CB burst 2
16	Reserved	Reserved	RW CB burst 1	RW CB burst 0
15:14	Reserved			
13:12	Reserved			
11:10	Read/Write Data burst 3 (Bytes 31:24)			
09:08	Read/Write Data burst 2 (Bytes 23:16)			
07:06	Reserved			
05:04	Reserved			
03:02	Read/Write Data burst 1 (Bytes 15:8)			
01:00	Read/Write Data burst 0 (Bytes 7:0)			

Table 400. Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0 (Sheet 6 of 7)

Offset: *108 - 10Bh* ***Size:*** *32 bits*

Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:00 (Cont'd)	DCAL Data	OPCODE: Memory Test (Data Type is Fixed Data Pattern). DCALDATA is used as an error log, the upper half logs the first 9 failing addresses and the lower half logs data bit lane failures. Faddr(31:0): a 32-bit value that is used to calculate the address of the failure. The address is captured at the time of the read comparison. Due to pipelining, the address advances numerous times before the error is detected. The lower 3 bits, Faddr(3:0), indicate how many address increments occurred before the error was detected. However, the address increment is not a simple "+1" operation. There is an address swizzle function before the "+1" operation, followed by an inverse swizzle function. The calculation of the failing address is a follows: Fail Address = unswizzle (swizzle (Faddr(31:3)) + Faddr(2:0))	0000_0000h	RW

DW(s)	31:24 (Byte 3)	23:16 (Byte 2)	15:8 (Byte 1)	7:0 (Byte 0)
17	Faddr8 Top 16 bits reserved			
16	Reserved	Reserved	CB Fail burst 3,1	CB Fail burst 2,0
15	Faddr7(31:0)			
14	Faddr6(31:0)			
13	Faddr5(31:0)			
12	Faddr4(31:0)			
11	Faddr3(31:0)			
10	Faddr2(31:0)			
09	Faddr1(31:0)			
08	Faddr0(31:0)			
07	Reserved			
06	Reserved			
05	Reserved			
04	Reserved			
03	DQ(63:32) Fail burst 3,1			
02	DQ(31:0) Fail burst 3,1			
01	DQ(63:32) Fail burst 2,0			
00	DQ(31:0) Fail burst 2,0			

Table 400. Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0 (Sheet 7 of 7)

Offset: 108 - 10Bh **Size:** 32 bits

Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:00 (Cont'd)	DCAL Data	OPCODE: Memory Test (Data Type is LFSR or Circular Swizzle). The upper half of DCALDATA is used to generate the write data and read compare data and the lower half of DCALDATA is used as an error log, which logs either the first 9 failing addresses (error log select = 1) or data bit lane failures (error log select = 0). Shown below is when error log select = 1 (address error log). DW(s) Description 17 gen_data_dw8 = gen_data_dw0(6:0,31:7) when Data Type = Circular Swizzle gen_data_dw8 = crc32_x4(gen_data_dw8) when Data Type = LFSR 16 Faddr8(31:0) 15 gen_data_dw7 = gen_data_dw8(30:0,31) 14 gen_data_dw6 = gen_data_dw7(30:0,31) 13 gen_data_dw5 = gen_data_dw6(30:0,31) 12 gen_data_dw4 = gen_data_dw5(30:0,31) 11 gen_data_dw3 = gen_data_dw4(30:0,31) 10 gen_data_dw2 = gen_data_dw3(30:0,31) 9 gen_data_dw1 = gen_data_dw2(30:0,31) 8 gen_data_dw0 = gen_data_dw1(30:0,31) 7 Faddr7(31:0) 6 Faddr6(31:0) 5 Faddr5(31:0) 4 Faddr4(31:0) 3 Faddr3(31:0) 2 Faddr2(31:0) 1 Faddr1(31:0) 0 Faddr0(31:0)	0000_0000h	RW
31:00 (Cont'd)	DCAL Data	OPCODE: Error Monitor, when no error monitor features are enabled; DCALCSR(7:4) = 0000. In this mode the error monitor reads a single read data FIFO buffer position from the DDR2 and stores this information in the following fields in DCALDATA.	0000_0000h	RW

DW(s)	31:24 (Byte 3)	23:16 (Byte 2)	15:8 (Byte 1)	7:0 (Byte 0)
17	Reserved (not used)			
16	Reserved	Reserved	CB(7:0) FIFO data burst 1	CB(7:0) FIFO data burst 0
15:08	Reserved (not used)			
07	Reserved			
06	Reserved			
05	Reserved			
04	Reserved			
03	DQ(63:32) FIFO burst 1			
02	DQ(31:0) FIFO burst 1			
01	DQ(63:32) FIFO burst 0			
00	DQ(31:0) FIFO burst 0			

13.7.1.6 Offset 10C - 10Fh: DCALDATA1 – DCAL Data Register DW1

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits, as with the DCAL Address Register, changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 401. Offset 10C - 10Fh: DCALDATA1 – DCAL Data Register DW1

Offset: *10C - 10Fh*		***Size:*** *32 bits*		
Default Value: *0000_0000h*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.7 Offset 110 - 113h: DCALDATA2 – DCAL Data Register DW2

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 402. Offset 110 - 113h: DCALDATA2 – DCAL Data Register DW2

Offset: *110 - 113h*		***Size:*** *32 bit*		
Default Value: *0000_0000h*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.8 Offset 114 - 117h: DCALDATA3 – DCAL Data Register DW3

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 403. Offset 114 - 117h: DCALDATA3 – DCAL Data Register DW3

Offset: *114 - 117h*		***Size:*** *32 bit*		
Default Value: *0000_0000h*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.9 Offset 118 - 11Bh: DCALDATA4 – DCAL Data Register DW4

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 404. Offset 118 - 11Bh: DCALDATA4 – DCAL Data Register DW4

Offset: 118 - 11Bh Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.10 Offset 11C - 11Fh: DCALDATA5 – DCAL Data Register DW5

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 405. Offset 11C - 11Fh: DCALDATA5 – DCAL Data Register DW5

Offset: 11C - 11Fh Default Value: 0000_0000h		Size: 32 bits		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.11 Offset 120 - 123h: DCALDATA6 – DCAL Data Register DW6

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 406. Offset 120 - 123h: DCALDATA6 – DCAL Data Register DW6

Offset: 120 - 123h Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.12 Offset 124 - 127h: DCALDATA7 – DCAL Data Register DW7

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 407. Offset 124 - 127h: DCALDATA7 – DCAL Data Register DW7

Offset: 124 - 127h Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.13 Offset 128 - 12Bh: DCALDATA8 – DCAL Data Register DW8

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 408. Offset 128 - 12Bh: DCALDATA8 – DCAL Data Register DW8

Offset: 128 - 12Bh Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.14 Offset 12C - 12Fh: DCALDATA9 – DCAL Data Register DW9

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 409. Offset 12C - 12Fh: DCALDATA9 – DCAL Data Register DW9

Offset: 12C - 12Fh Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.15 Offset 130 - 133h: DCALDATA10 – DCAL Data Register DW10

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 410. Offset 130 - 133h: DCALDATA10 – DCAL Data Register DW10

Offset: 130 - 133h Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.16 Offset 134 - 137h: DCALDATA11 – DCAL Data Register DW11

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 411. Offset 134 - 137h: DCALDATA11 – DCAL Data Register DW11

Offset: 134 - 137h Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.17 Offset 138 - 13Bh: DCALDATA12 – DCAL Data Register DW12

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 412. Offset 138 - 13Bh: DCALDATA12 – DCAL Data Register DW12

Offset: 138 - 13Bh Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.18 Offset 13C - 13Fh: DCALDATA13 – DCAL Data Register DW13

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 413. Offset 13C - 13Fh: DCALDATA13 – DCAL Data Register DW13

Offset: 13C - 13Fh Default Value: 0000_0000h		Size: 32 bits		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.19 Offset 140 - 143h: DCALDATA14 – DCAL Data Register DW14

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 414. Offset 140 - 143h: DCALDATA14 – DCAL Data Register DW14

Offset: 140 - 143h Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.20 Offset 144 - 147h: DCALDATA15 – DCAL Data Register DW15

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 415. Offset 144 - 147h: DCALDATA15 – DCAL Data Register DW15

Offset: 144 - 147h Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.21 Offset 148 - 14Bh: DCALDATA16 – DCAL Data Register DW16

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 416. Offset 148 - 14Bh: DCALDATA16 – DCAL Data Register DW16

Offset: 148 - 14Bh Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.22 Offset 14C - 14Fh: DCALDATA17 – DCAL Data Register DW17

This 32-bit register is one of 18 data registers used to support the various DCAL opcodes. The definition of these bits as with the DCAL Address Register changes depending on the operation. See Section 13.7.1.5, "Offset 108 - 10Bh: DCALDATA0 – DCAL Data Register DW0" for register field definitions.

Table 417. Offset 14C - 14Fh: DCALDATA17 – DCAL Data Register DW17

Offset: 14C - 14Fh Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DCAL Data	The definition is opcode dependent.	0	RW

13.7.1.23 Offset 150 - 153h: RCVENDLYA – Receive Enable Delay Register

The IMCH implementation supports a single timing value per DIMM that is used to determine when to enable the DQS input receiver. The IMCH relies on an internally generated "receive-enable" to enable sampling of the DQS strobes, and the fields of this register encode the delay from read command to the center of the DQS strobe pre-amble. A designated CS pair refer to the logical CS pair before the DRM register.

Table 418. Offset 150 - 153h: RCVENDLYA – Receive Enable Delay Register

Offset: 150 - 153h Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:29	Reserved	Reserved	000b	
28:24	REDCS6	**Receive Enable Delay for CS 6**	00h	RW
23:21	Reserved	Reserved	000b	
20:16	REDCS4	**Receive Enable Delay for CS 4**	00h	RW
15:13	Reserved	Reserved	000b	
12:08	REDCS2	**Receive Enable Delay for CS 2**	00h	RW
07:05	Reserved	Reserved	000b	
04:00	REDCS0	**Receive Enable Delay for CS 0:** 04:03 select the target DDR2 clock (0-3) for the center of the pre-amble 02:00 select the offset (0-7) into the target DDR2 clock in 1/8th clock increments	00h	RW

13.7.1.24 Offset 200 - 203h: DQSOFCSA01L – DQS Offset to CS0&1 Lower Register

This 32-bit register consists of the first 8 of 18 packed 4-bit fields for DQS offset to CS0&1.

Table 419. Offset 200 - 203h: DQSOFCSA01L – DQS Offset to CS0&1 Lower Register

Offset: *200 - 203h* ***Size:*** *32 bit*

Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:28	OFFDQS7	**Offset for DQS7:** Calibration is performed by the DCAL engine to determine how the DQS must be adjusted so that the incoming DQ is clocked in the middle of the data valid window. While the coarse settings for the DLL circuitry can be made based on frequency alone, there is still some variability, due to such things as signal path length, that the calibration process determines. BIOS loads this nibble value based on the outcome of the calibration process. This fine granularity value offsets the positioning of DQS from a fixed starting point based on the coarse settings.	0000b	RW
27:24	OFFDQS6	**Offset for DQS6:** Same field definition as for DQS7.	0000b	RW
23:20	OFFDQS5	**Offset for DQS5:** Same field definition as for DQS7.	0000b	RW
19:16	OFFDQS4	**Offset for DQS4:** Same field definition as for DQS7.	0000b	RW
15:12	OFFDQS3	**Offset for DQS3:** Same field definition as for DQS7.	0000b	RW
11:08	OFFDQS2	**Offset for DQS2:** Same field definition as for DQS7.	0000b	RW
07:04	OFFDQS1	**Offset for DQS1:** Same field definition as for DQS7.	0000b	RW
03:00	OFFDQS0	**Offset for DQS0:** Same field definition as for DQS7.	0000b	RW

13.7.1.25 Offset 204 - 207h: DQSOFCSA01M – DQS Offset to CS0&1 Middle Register

This 32-bit register consists of the second 8 of 18 packed 4-bit fields for DQS offset CS0&1.

Table 420. Offset 204 - 207h: DQSOFCSA01M – DQS Offset to CS0&1 Middle Register

Offset: *204 - 207h* ***Size:*** *32 bit*

Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:28	OFFDQS15	**Offset for DQS15:** Refer to Section 13.7.1.24, "Offset 200 - 203h: DQSOFCSA01L – DQS Offset to CS0&1 Lower Register" for a brief description.	0000b	RW
27:24	OFFDQS14	**Offset for DQS14:** Same field definition as for DQS15.	0000b	RW
23:20	OFFDQS13	**Offset for DQS13:** Same field definition as for DQS15.	0000b	RW
19:16	OFFDQS12	**Offset for DQS12:** Same field definition as for DQS15.	0000b	RW
15:12	OFFDQS11	**Offset for DQS11:** Same field definition as for DQS15.	0000b	RW
11:08	OFFDQS10	**Offset for DQS10:** Same field definition as for DQS15.	0000b	RW
07:04	OFFDQS9	**Offset for DQS9:** Same field definition as for DQS15.	0000b	RW
03:00	OFFDQS8	**Offset for DQS8:** Same field definition as for DQS15.	0000b	RW

13.7.1.26 Offset 208h: DQSOFCSA01U – DQS Offset to CS0&1 Upper Register

This 8-bit register consists of the last two of 18 packed 4-bit fields for DQS offset to CS0&1.

Table 421. Offset 208h: DQSOFCSA01U – DQS Offset to CS0&1 Upper Register

Offset: 208h		*Size:* 8-bit		
Default Value: 00h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:04	OFFDQS17	**Offset for DQS17:** Refer to Section 13.7.1.24, "Offset 200 - 203h: DQSOFCSA01L – DQS Offset to CS0&1 Lower Register" for a brief description.	0000b	RW
03:00	OFFDQS16	**Offset for DQS16:** Same field definition as for DQS17.	0000b	RW

13.7.1.27 Offset 20C - 20Fh: DQSOFCSA23L – DQS Offset to CS2&3 Lower Register

This 32-bit register consists of the first eight of 18 packed 4-bit fields for DQS offset to CS1.

Table 422. Offset 20C - 20Fh: DQSOFCSA23L – DQS Offset to CS2&3 Lower Register

Offset: 20C - 20Fh		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.7.1.24, "Offset 200 - 203h: DQSOFCSA01L – DQS Offset to CS0&1 Lower Register" for bit definitions.				

13.7.1.28 Offset 210 - 213h: DQSOFCSA23M – DQS Offset to Channel A CS2&3 Middle Register

This 32-bit register consists of the second eight of 18 packed 4-bit fields for DQS offset to CS2&3.

Table 423. Offset 210 - 213h: DQSOFCSA23M – DQS Offset to Channel A CS2&3 Middle Register

Offset: 210 - 213h		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.7.1.25, "Offset 204 - 207h: DQSOFCSA01M – DQS Offset to CS0&1 Middle Register" for bit definitions.				

13.7.1.29 Offset 214h: DQSOFCSA23U – DQS Offset to CS2&3 Upper Register

This 8-bit register consists of the last two of 18 packed 4-bit fields for DQS offset to CS2&3.

Table 424. Offset 214h: DQSOFCSA23U – DQS Offset to CS2&3 Upper Register

Offset: 214h		Size: 8-bit		
Default Value: 00h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.7.1.26, "Offset 208h: DQSOFCSA01U – DQS Offset to CS0&1 Upper Register" for bit definitions.				

13.7.1.30 Offset 218 - 21Bh: DQSOFCSA45L – DQS Offset to CS4&5 Lower Register

This 32-bit register consists of the first eight of 18 packed 4-bit fields for DQS offset to CS2&3.

Table 425. Offset 218 - 21Bh: DQSOFCSA45L – DQS Offset to CS4&5 Lower Register

Offset: 218 - 21Bh		Size: 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.7.1.24, "Offset 200 - 203h: DQSOFCSA01L – DQS Offset to CS0&1 Lower Register" for bit definitions.				

13.7.1.31 Offset 21C - 21Fh: DQSOFCSA45M – DQS Offset to CS4&5 Middle Register

This 32-bit register consists of the second eight of 18 packed 4-bit fields for DQS offset to CS3.

Table 426. Offset 21C - 21Fh: DQSOFCSA45M – DQS Offset to CS4&5 Middle Register

Offset: 21C - 21Fh		Size: 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.7.1.25, "Offset 204 - 207h: DQSOFCSA01M – DQS Offset to CS0&1 Middle Register" for bit definitions.				

13.7.1.32 Offset 220h: DQSOFCSA45U – DQS Offset to CS4&5 Upper Register

This 8-bit register consists of the last two of 18 packed 4-bit fields for DQS offset to CS4.

Table 427. Offset 220h: DQSOFCSA45U – DQS Offset to CS4&5 Upper Register

Offset: 220h		Size: 8-bit		
Default Value: 00h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.7.1.26, "Offset 208h: DQSOFCSA01U – DQS Offset to CS0&1 Upper Register" for bit definitions.				

13.7.1.33 Offset 224 - 227h: DQSOFCSA67L – DQS Offset to A CS6&7 Lower Register

This 32-bit register consists of the first eight of 18 packed 4-bit fields for DQS offset to CS6&7.

Table 428. Offset 224 - 227h: DQSOFCSA67L – DQS Offset to A CS6&7 Lower Register

Offset: *224 - 227h* ***Size:*** *32 bit* ***Default Value:*** *0000_0000h*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.7.1.24, "Offset 200 - 203h: DQSOFCSA01L – DQS Offset to CS0&1 Lower Register" for bit definitions.				

13.7.1.34 Offset 228 - 22Bh: DQSOFCSA67M – DQS Offset to CS6&7 Middle Register

This 32-bit register consists of the second eight of 18 packed 4-bit fields for DQS offset to CS6&7.

Table 429. Offset 228 - 22Bh: DQSOFCSA67M – DQS Offset to CS6&7 Middle Register

Offset: *228 - 22Bh* ***Size:*** *32 bit* ***Default Value:*** *0000_0000h*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.7.1.25, "Offset 204 - 207h: DQSOFCSA01M – DQS Offset to CS0&1 Middle Register" for bit definitions.				

13.7.1.35 Offset 22Ch: DQSOFCSA67U – DQS Offset to CS6&7 Upper Register

This 8-bit register consists of the last two of 18 packed 4-bit fields for DQS offset to CS6&7.

Table 430. Offset 22Ch: DQSOFCSA67U – DQS Offset to CS6&7 Upper Register

Offset: *22Ch* ***Size:*** *8-bit* ***Default Value:*** *00h*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
See Section 13.7.1.26, "Offset 208h: DQSOFCSA01U – DQS Offset to CS0&1 Upper Register" for bit definitions.				

13.8 Memory Mapped I/O for EDMA Registers

This section describes the memory-mapped registers for the EDMA Controller. The EDMALBAR register, described in Section 13.3.1.9, "Offset 10h - 13h: EDMALBAR – EDMA Low Base Address Register" provides the base address for these registers. The offsets listed for the following registers are relative to this base address.

The BAR value for all registers in this section is BAR10h.

Each EDMA channel consists of twelve 32-bit registers contiguous in memory mapped address space. The first of the four sets is described in detail, the others are copies at different offsets for the other channels. The abbreviations for the other channel register names replace the 0 at the end of the name with the appropriate channel number 1, 2, or 3.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 431. Memory Mapped I/O for EDMA Register Summary (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Sticky	Default	Access
Start	End					
Memory Mapped I/O for Channel 0						
00h	03h	CCR0	Channel Control Register	No	0000_0000h	RW
04h	07h	CSR0	Channel Status Register	No	0000_0000h	RO, RWC
08h	0Bh	CDAR0	Current Descriptor Address Register	No	0000_0000h	RO
0Ch	0Fh	CDUAR0	Current Descriptor Upper Address Register	No	0000_0000h	RO
10h	13h	SAR0	Source Address Register	No	0000_0000h	RO
14h	17h	SUAR0	Source Upper Address Register	No	0000_0000h	RO
18h	1Bh	DAR0	Destination Address Register	No	0000_0000h	RO
1Ch	1Fh	DUAR0	Destination Upper Address Register	No	0000_0000h	RO
20h	23h	NDAR0	Next Descriptor Address Register	No	0000_0000h	RWL
24h	27h	NDUAR0	Next Descriptor Upper Address Register	No	0000_0000h	RWL
28h	2Bh	TCR0	Transfer Count Register	No	0000_0000h	RO
2Ch	2Fh	DCR0	Descriptor Control Register	No	0000_0000h	RO
Memory Mapped I/O for Channel 1						
40h	43h	CCR1	Channel Control Register	No	0000_0000h	RO, RW, RWS
44h	47h	CSR1	Channel Status Register	No	0000_0000h	RO, RWC
48h	4Bh	CDAR1	Current Descriptor Address Register	No	0000_0000h	RO
4Ch	4Fh	CDUAR1	Current Descriptor Upper Address Register	No	0000_0000h	RO
50h	53h	SAR1	Source Address Register	No	0000_0000h	RO
54h	57h	SUAR1	Source Upper Address Register	No	0000_0000h	RO
58h	5Bh	DAR1	Destination Address Register	No	0000_0000h	RO
5Ch	5Fh	DUAR1	Destination Upper Address Register	No	0000_0000h	RO
60h	63h	NDAR1	Next Descriptor Address Register	No	0000_0000h	RWL
64h	67h	NDUAR1	Next Descriptor Upper Address Register	No	0000_0000h	RWL
68 h	6Bh	TCR1	Transfer Count Register	No	0000_0000h	RO
6Ch	6Fh	DCR1	Descriptor Control Register	No	0000_0000h	RO
Memory Mapped I/O for Channel 2						
80h	83h	CCR2	Channel Control Register	No	0000_0000h	RO, RW, RWS
84h	87h	CSR2	Channel Status Register	No	0000_0000h	RO, RWC

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 431. Memory Mapped I/O for EDMA Register Summary (Sheet 2 of 2)

Offset		Symbol	Register Name/Function	Sticky	Default	Access
Start	End					
88h	8Bh	CDAR2	Current Descriptor Address Register	No	0000_0000h	RO
8Ch	8Fh	CDUAR2	Current Descriptor Upper Address Register	No	0000_0000h	RO
90h	93h	SAR2	Source Address Register	No	0000_0000h	RO
94h	97h	SUAR2	Source Upper Address Register	No	0000_0000h	RO
98h	9Bh	DAR2	Destination Address Register	No	0000_0000h	RO
9Ch	9Fh	DUAR2	Destination Upper Address Register	No	0000_0000h	RO
A0h	A3h	NDAR2	Next Descriptor Address Register	No	0000_0000h	RWL
A4h	A7h	NDUAR2	Next Descriptor Upper Address Register	No	0000_0000h	RWL
A8h	ABh	TCR2	Transfer Count Register	No	0000_0000h	RO
ACh	AFh	DCR2	Descriptor Control Register	No	0000_0000h	RO
Memory Mapped I/O for Channel 3						
C0h	C3h	CCR3	Channel Control Register	No	0000_0000h	RO, RW, RWS
C4h	C7h	CSR3	Channel Status Register	No	0000_0000h	RO, RWC
C8h	CBh	CDAR3	Current Descriptor Address Register	No	0000_0000h	RO
CCh	CFh	CDUAR3	Current Descriptor Upper Address Register	No	0000_0000h	RO
D0h	D3h	SAR3	Source Address Register	No	0000_0000h	RO
D4h	D7h	SUAR3	Source Upper Address Register	No	0000_0000h	RO
D8h	DBh	DAR3	Destination Address Register	No	0000_0000h	RO
DCh	DFh	DUAR3	Destination Upper Address Register	No	0000_0000h	RO
E0h	E3h	NDAR3	Next Descriptor Address Register	No	0000_0000h	RWL
E4h	E7h	NDUAR3	Next Descriptor Upper Address Register	No	0000_0000h	RWL
E8h	EBh	TCR3	Transfer Count Register	No	0000_0000h	RO
ECh	EFh	DCR3	Descriptor Control Register	No	0000_0000h	RO
Memory Mapped I/O for the EDMA Controller						
100h	103h	DCGC	EDMA Controller Global Command Register	No	0000_0000h	RO, RW
104h	107h	DCGS	EDMA Controller Global Status Register	No	0000_0000h	RO

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

13.8.1 Register Details

13.8.1.1 Offset 00 - 03h: CCR0 – Channel 0 Channel Control Register

The Channel Control Register (CCR) is cleared to zero on power-on or system reset. The CCR specifies the overall operating environment for the channel. Software initializes this register only after initializing the chain descriptors in system memory, and the Next Address registers as pointer to the first chain descriptor in memory. CCR can be written when the EDMA channel is active. The CCR is a read/write register.

Table 432. Offset 00 - 03h: CCR0 – Channel 0 Channel Control Register

Offset: 00 - 03h **Size:** 32 bit
Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
31:04	Reserved	Reserved	000000h	
03	CRSM	Channel Resume: 0 = Cleared when: • The channel completes a EDMA transfer and the Next Descriptor Address Register is not zero. In this case the channel proceeds to the next descriptor in the chain (resumes). • The channel is idle or the channel completes a EDMA transfer and the Next Descriptor Address Register is zero. 1 = Causes the channel to resume chaining by re-reading the current descriptor located in local system memory and reloading the Next Descriptor Address Register when the channel is idle (the Channel Active bit in the CSR is clear) or when the channel completes execution of the current descriptor. Once set, software cannot clear this bit. The IMCH prevents this bit from being set when either the stopped or aborted bit is set in the CSR. Software must clear the CSR stopped and aborted bits before attempting to resume the current descriptor chain. If the CSR end of chain bit was set, the EDMA channel clears the end of chain bit when the current descriptor chain resumes. Refer to "Enhanced Direct Memory Access Controller (EDMA)" on page 110 for details on the suspend and resume function.	0b	RWS
02	STPDMA	**Stop:** 0 = Cleared only by the IMCH, once the Channel Active bit is cleared and the EDMA Stopped bit is set. 1 = Causes the current EDMA transfer to stop. The channel does not request the bus on the source side. Any data in the queue is emptied to the destination side, and all relevant bits in the CCR (bits 03:00) and CSR (Channel Active bit) are cleared. This bit has priority over the Suspend EDMA bit. Once set, this bit cannot be cleared by the software. Software must be very careful in setting this bit since any EDMA transfer, once stopped, cannot be restarted from that point.	0b	RWS
01	SUSDMA	**Suspend:** This has no effect on the Channel Active bit. 0 = Software clears this bit once the EDMA Suspended bit is set. Clearing this bit restarts the EDMA transfer from the point it was suspended, and clears the EDMA Suspended bit in the CSR. Refer to "Enhanced Direct Memory Access Controller (EDMA)" on page 110 of for details on the EDMA suspend function. 1 = Allows the current descriptor to finish, but suspends channel chaining. The channel continues to request the bus on the source side for the current descriptor. When the data in the queue for this descriptor is emptied to the destination side, the channel sets the EDMA Suspended bit in the CSR.	0b	RW
00	STRTDMA	**Start:** 0 = Cleared by the IMCH when the EDMA transfer is complete, when the EDMA is stopped by software, or when the EDMA encounters any unrecoverable error. The IMCH prevents this bit from being set when the stopped or aborted bit is set in the CSR. The EDMA channel must be idle and software must clear the CSR before starting the EDMA channel with a new descriptor chain. 1 = Channel is enabled for EDMA transfer. Once set, this bit cannot be cleared by software.	0b	RWS

13.8.1.2 Offset 04 - 07h: CSR0 – Channel 0 Channel Status Register

The Channel Status Register (CSR) contains status flags that indicate the channel status. The register is read by application software to get the current channel status and to examine the source of an interrupt. Table 433 shows the format for the CSR.

Table 433. Offset 04 - 07h: CSR0 – Channel 0 Channel Status Register

Offset: *04 - 07h* ***Size:*** *32 bit*

Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:06	Reserved	Reserved	0000000h	
05	CACTV	**Channel Active:** 0 = Channel is inactive and available to be configured for EDMA transfer by software. 1 = Set by the IMCH, indicates the channel is in use and actively performing EDMA data transfers. The channel active flag is set by the IMCH when: • Software initiates a EDMA transfer by setting the Start bit of CCR and the EDMA channel in response loads the chain descriptor from the local system memory • Software initiates a EDMA transfer by setting the Channel Resume bit of the CCR and the NDAR/NDUAR point to a legal non-null address in memory.	0b	RO
04	DABRT	**Aborted:** 0 = Software clears this bit by writing a 1 to the bit location. 1 = Indicates that the current EDMA transfer for this channel encountered an unrecoverable error. If the Aborted Interrupt Enable bit in the DCR is set, this generates an interrupt to the processor. Software polls this bit if an interrupt is not enabled. Error details are logged in the DMA_FERR and DMA_NERR registers.	0b	RWC
03	DSTP	**Stopped:** 0 = Software clears this bit by writing a 1 to the bit location. 1 = Indicates that the current transfer for this channel has been stopped by software setting the Stop bit in the CCR. If the Stopped Interrupt Enable bit in the DCR is set, this generates an interrupt to the processor. Software can use this bit for polling if interrupts are not enabled.	0b	RWC
02	DSUS	**Suspended:** 0 = Cleared when software clears the Suspend bit in the CCR. 1 = Indicates that the current transfer for this channel has been stopped by software setting the Suspend bit in the CCR. If the Suspended Interrupt Enable bit in the DCR is set, this generates an interrupt to the processor. Software can use this bit for polling if interrupts are not enabled.	0b	RO
01	EOT	**End of Transfer:** 0 = Software clears this bit by writing a ‘1’ to the bit location. 1 = Indicates that the channel has successfully completed an error-free EDMA transfer of at least one descriptor. If the End of Transfer Interrupt Enable bit in the DCR is set, this generates an interrupt to the processor. Software can use this bit for polling if interrupts are not enabled.	0b	RWC
00	EOC	**End of Chain:** 0 = Software clears this bit by writing a ‘1’ to the bit location. 1 = Indicates that the channel has successfully completed an error-free EDMA transfer, and it is the last descriptor in a chain descriptor. If the End of Chain Interrupt Enable bit in the DCR is set, this generates an interrupt to the processor. Software can use this bit for polling if interrupts are not enabled.	0b	RWC

13.8.1.3 Offset 08 - 0Bh: CDAR0 – Channel 0 Current Descriptor Address Register

The Current Descriptor Address Register (CDAR) contains the lower 32-bit address of the current chain descriptor in local system memory. This register is loaded by the IMCH when a new chain descriptor is read. All chain descriptors are aligned on an eight double-word (32 bit) boundary.

Table 434. Offset 08 - 0Bh: CDAR0 – Channel 0 Current Descriptor Address Register

Offset: 08 - 0Bh		*Size:* 32 bit		
Default Value: 0000_0000h		*Power Well:*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:05	CDADD	**Current Descriptor Address:** Lower 32 bits of the local system memory address of the current chain descriptor that is read by the channel. The descriptor address must be eight double-word aligned.	0000000h	RO
04:00	Reserved	Reserved.	00h	

13.8.1.4 Offset 0C - 0Fh: CDUAR0–Channel 0 Current Descriptor Upper Address Register

The Current Descriptor Upper Address Register (CDUAR) contains the upper 32-bit address of the current chain descriptor in local system memory. This register is loaded by the IMCH when a new chain descriptor is read.

Table 435. Offset 0C - 0Fh: CDUAR0–Channel 0 Current Descriptor Upper Address Register

Offset: 0C - 0Fh		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	CDUAR0	**Current Descriptor Address:** The upper 32-bit local system memory address of the current chain descriptor that is read by the channel.	0000000h	RO

13.8.1.5 Offset 10 - 13h: SAR0 – Channel 0 Source Address Register

The Source Address Register (SAR) contains the lower 32-bit source address for the current EDMA transfer. This register is loaded by the IMCH when the source address field of a new chain descriptor is read.

Table 436. Offset 10 - 13h: SAR0 – Channel 0 Source Address Register

Offset: 10 - 13h		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	SAR0	**Current Source Address:** The lower 32-bit source memory address for the current EDMA transfer.	0000000h	RO

13.8.1.6 Offset 14 - 17h: SUAR0 – Channel 0 Source Upper Address Register

The Source Upper Address Register (SUAR) contains the upper 32-bit source address for the current EDMA transfer. This register is loaded by the IMCH when the source upper address field of a new chain descriptor is read.

Table 437. Offset 14 - 17h: SUAR0 – Channel 0 Source Upper Address Register

Offset: 14 - 17h		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	SUAR0	**Current Source Address:** The upper 32-bit source memory address for the current EDMA transfer.	0000000h	RO

13.8.1.7 Offset 18 - 1Bh: DAR0 – Channel 0 Destination Address Register

The Destination Address Register (DAR) contains the lower 32-bit destination address for the current EDMA transfer. This register is loaded by the IMCH when the destination address field of a new chain descriptor is read.

Table 438. Offset 18 - 1Bh: DAR0 – Channel 0 Destination Address Register

Offset: 18 - 1Bh		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DAR0	**Current Destination Address:** The lower 32-bit destination memory address for the current EDMA transfer.	0000000h	RO

13.8.1.8 Offset 1C - 1Fh: DUAR0 – Channel 0 Destination Upper Address Register

The Destination Upper Address Register (DUAR) contains the upper 32-bit destination address for the current EDMA transfer. This register is loaded by the IMCH when the destination upper address field of a new chain descriptor is read.

Table 439. Offset 1C - 1Fh: DUAR0 – Channel 0 Destination Upper Address Register

Offset: 1C - 1Fh		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DUAR0	**Current Destination Address:** The upper 32-bit destination memory address for the current EDMA transfer.	0000000h	RO

13.8.1.9 Offset 20 - 23h: NDAR0 – Channel 0 Next Descriptor Address Register

The Next Descriptor Address Register (NDAR) contains the lower 32-bit address of the next descriptor chain in the local system memory. This register is loaded when the next descriptor address field of a new chain descriptor is read. Additionally, software writes this register with the address of the first chain descriptor in local memory. All chain descriptors are required to be aligned on an eight double-word (32-bit) boundary or Intel® 3100 Chipset flags an error.

Note: Software must make sure that the Start bit in the CCR and the Channel Active bit in the CSR are clear prior to writing to the NDAR. The IMCH prevents writing to this register when these bits are not clear. Writing zero into the NDAR and NDUAR by software does not start a EDMA transfer.

Table 440. Offset 20 - 23h: NDAR0 – Channel 0 Next Descriptor Address Register

Offset: *20 - 23h* ***Default Value:*** *0000_0000h*		***Size:*** *32 bit*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	NDLADD	**Next Descriptor Lower Address:** Lower 32 bits of the local system memory address of the next chain descriptor in memory to be read by the channel. The address must be aligned on an eight DWord (32-bit) boundary or else the IMCH flags an error. This field can only be written when the Start bit in the CCR and the Channel Active bit in the CSR are clear.	0000000h	RWL

13.8.1.10 Offset 24 - 27h: NDUAR0 – Channel 0 Next Descriptor Upper Address Register

The Next Descriptor Upper Address Register (NDUAR) contains the upper 32-bit address of the next descriptor chain in the local system memory. This register is loaded when the next descriptor address field of a new chain descriptor is read. Additionally, software writes this register with the address of the first chain descriptor in local memory.

Note: Software must make sure that the Start bit in the CCR and the Channel Active bit in the CSR are clear prior to writing to the Next Descriptor Upper Address Register (NDAR). The IMCH prevents writing to this register when these bits are not clear. Writing zero into the NDAR and NDUAR by application software does not start a EDMA transfer.

Table 441. Offset 24 - 27h: NDUAR0 – Channel 0 Next Descriptor Upper Address Register

Offset: *24 - 27h* ***Default Value:*** *0000_0000h*		***Size:*** *32 bit*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	NDUADD	**Next Descriptor Upper Address:** The upper 32-bit address of the next descriptor chain in memory to be read by the channel. This field can only be written when the Start bit in the CCR and the Channel Active bit in the CSR are clear.	0000000h	RWL

13.8.1.11 Offset 28 - 2Bh: TCR0 – Channel 0 Transfer Count Register

The Transfer Count Register (TCR) contains the number of bytes it transfers. This register is loaded when the transfer count field of the chain descriptor is read from memory. The maximum allowed value for the TCR is 16 Mbytes. Values greater than 16 Mbytes are truncated to 16 Mbytes and no error is reported. A value of zero is valid and results in no data being transferred and no cycles are generated on the source or destination buses. During transfers, this register contains the remaining byte bytes to be written to the destination.

Table 442. Offset 28 - 2Bh: TCR0 – Channel 0 Transfer Count Register

Offset: 28 - 2Bh Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:25	Reserved	Reserved	0000000h	
24:00	TCR0	**Transfer Count:** Set by the IMCH when the transfer count field of the chain descriptor is read from memory. It reflects the number of bytes for a EDMA transfer. A value of 0 results in no data being transferred. The maximum value that can be programmed to this register is 16 Mbytes. Larger values written in the transfer count field of the chain descriptor are truncated, and no error is reported. Refer to DCR0[18:17] for additional programming requirements when in Constant Address Mode.	000000h	RO

13.8.1.12 Offset 2C - 2Fh: DCR0 – Channel 0 Descriptor Control Register

The Descriptor Control Register (DCR) contains control values for the EDMA transfer on a per descriptor basis. This register is loaded when the descriptor control field of the chain descriptor is read from memory. The value for this register may vary from chain descriptor to chain descriptor.

Table 443. Offset 2C - 2Fh: DCR0 – Channel 0 Descriptor Control Register (Sheet 1 of 3)

Offset: 2C - 2Fh Default Value: 0000_0000h		Size: 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:29	TC	PCI Express A-segment Traffic Class: This field is used to set the Traffic Class field on the PCI Express bus for transactions. This field has no effect on memory to memory transactions.	000b	RO
28:19	Reserved	Reserved	000h	
18:17	Granularity	**Destination Granularity:** 00 1 byte granularity 01 2 byte granularity (DAR[0] and TCR[0] ignored) 10 4 byte granularity (DAR[1:0] and TCR[1:0] ignored) 11 Reserved This bit are loaded by the descriptor fetch only. This field is ignored unless bits 15:14 are 01b (selecting Constant Destination Mode). When this field is enabled, the Destination Address and Transfer Count Register must contain an integer multiple of the granularity.	00b	RO
16	Reserved	Reserved	0b	
15:14	DADDM	**Destination Processing Mode:** 00 Increment mode 01 Constant mode (bits 18:17 set the granularity) 10 Reserved 11 Reserved These bits are loaded by the descriptor fetch only.	00b	RO

Table 443. Offset 2C - 2Fh: DCR0 – Channel 0 Descriptor Control Register (Sheet 2 of 3)

Offset: 2C - 2Fh **Size:** 32 bit

Default Value: 0000_0000h

Bits	Name	Description	Reset Value	Access
13:12	SADDM	**Source Processing Mode:** 00 Increment mode 01 Decrement mode 10 Buffer Initialization 11 Reserved These bits are loaded by the descriptor fetch only.	00b	RO
11:9	Reserved	Reserved	000b	
08	SRCC	**Source Coherency:** 0 = Source is a non-coherent address space 1 = Source is a coherent address space	0b	RO
07	DSTC	**Destination Coherency:** 0 = Destination is a non-coherent address space 1 = Destination is a coherent address space For PCI-E writes (I/O), this bit inversely reflects the state of the Snoop Not Required Attribute header bit: 0 = Snoop not required attribute bit = 1 1 = Snoop not required attribute bit = 0	0b	RO
06	SRCT	**Source Type:** Hardwired to 0: 0 = Indicates the source address points to local system memory. 1 = Indicates that the source address points to I/O memory. Although an I/O source type is not supported, this bit will reflect the actual source type as written by software. The IMCH flags an error if the source address range is not valid, and ignores the value of this bit.	0b	RO
05	DSTT	**Destination Type:** 0 = Destination Address points to local system memory 1 = Destination Address points to I/O space An error is signaled by the IMCHif the Destination Address type read from the descriptor does not match this setting.	0b	RO
04	DABRTIE	**Aborted Interrupt Enable:** Indicates whether or not an interrupt is generated when the EDMA Aborted (DABRT) bit in the CSR is set. 0 = Disable 1 = Enable	0b	RO
03	DSTPIE	**Stopped Interrupt Enable:** Indicates whether or not an interrupt is generated when the EDMA Stopped (DSTP) bit in the CSR is set. 0 = Disable 1 = Enable	0b	RO

Table 443. Offset 2C - 2Fh: DCR0 – Channel 0 Descriptor Control Register (Sheet 3 of 3)

Offset: *2C - 2Fh* ***Default Value:*** *0000_0000h*		***Size:*** *32 bit*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
02	DSUSIE	**Suspended Interrupt Enable:** Indicates whether or not an interrupt is generated when the EDMA Suspended (DSUS) bit in the CSR is set. 0 = Disable 1 = Enable	0b	RO
01	EOTIE	**End of Transfer Interrupt Enable:** Indicates whether or not an interrupt is generated when the End of Transfer (EOT) bit in the CSR is set. 0 = Disable 1 = Enable	0b	RO
00	EOCIE	**End of Chain Interrupt Enable:** Indicates whether or not an interrupt is generated when the End of Chain (EOC) bit in the CSR is set. 0 = Disable 1 = Enable	0b	RO

13.8.1.13 Offset 40 - 43h: CCR1 – Channel 1 Channel Control Register

Table 444. Offset 40 - 43h: CCR1 – Channel 1 Channel Control Register

Offset: *40 - 43h* ***Default Value:*** *0000_0000h*		***Size:*** *32 bit*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those described for CCR0 in Section 13.8.1.1.				

13.8.1.14 Offset 44 - 47h: CSR1 – Channel 1 Channel Status Register

Table 445. Offset 44 - 47h: CSR1 – Channel 1 Channel Status Register

Offset: *44 - 47h* ***Default Value:*** *0000_0000h*		***Size:*** *32 bit*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those described for CSR0 in Section 13.8.1.2.				

13.8.1.15 Offset 48 - 4Bh: CDAR1 – Channel 1 Current Descriptor Address Register

Table 446. Offset 48 - 4Bh: CDAR1 – Channel 1 Current Descriptor Address Register

Offset: 48 - 4Bh		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those described for CDAR0 in Section 13.8.1.3.				

13.8.1.16 Offset 4C - 4Fh: CDUAR1 – Channel 1 Current Descriptor Upper Address Register

Table 447. Offset 4C - 4Fh: CDUAR1 – Channel 1 Current Descriptor Upper Address Register

Offset: 4C - 4Fh		*Size:* 32 bit		
BAR: BAR10h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those described for CDUAR0 in Section 13.8.1.3.				

13.8.1.17 Offset 50 - 53h: SAR1 – Channel 1 Source Address Register

Table 448. Offset 50 - 53h: SAR1 – Channel 1 Source Address Register

Offset: 50 - 53h		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those described for SAR0 in Section 13.8.1.5.				

13.8.1.18 Offset 54 - 57h: SUAR1 – Channel 1 Source Upper Address Register

Table 449. Offset 54 - 57h: SUAR1 – Channel 1 Source Upper Address Register

Offset: 54 - 57h		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for SUAR0 described in Section 13.8.1.6.				

13.8.1.19 Offset 58 - 5Bh: DAR1 – Channel 1 Destination Address Register

Table 450. Offset 58 - 5Bh: DAR1 – Channel 1 Destination Address Register

Offset: 58 - 5Bh		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those described for DAR0 in Section 13.8.1.7.				

13.8.1.20 Offset 5C - 5Fh: DUAR1 – Channel 1 Destination Upper Address Register

Table 451. Offset 5C - 5Fh: DUAR1 – Channel 1 Destination Upper Address Register

Offset: 5C - 5Fh		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for DUAR0 described in Section 13.8.1.8.				

13.8.1.21 Offset 60 - 63h: NDAR1 – Channel 1 Next Descriptor Address Register

Table 452. Offset 60 - 63h: NDAR1 – Channel 1 Next Descriptor Address Register

Offset: 60 - 63h		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for NDAR0 described in Section 13.8.1.9.				

13.8.1.22 Offset 64 - 67h: NDUAR1 – Channel 1 Next Descriptor Upper Address Register

Table 453. Offset 64 - 67h: NDUAR1 – Channel 1 Next Descriptor Upper Address Register

Offset: 64 - 67h		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for NDUAR0 described in Section 13.8.1.10.				

13.8.1.23 Offset 68 - 6Bh: TCR1 – Channel 1 Transfer Count Register

Table 454. Offset 68 - 6Bh: TCR1 – Channel 1 Transfer Count Register

Offset: 68 - 6Bh *Default Value:* 0000_0000h		*Size:* 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for TCR0 described in Section 13.8.1.11.				

13.8.1.24 Offset 6C - 6Fh: DCR1 – Channel 1 Descriptor Control Register

Table 455. Offset 6C - 6Fh: DCR1 – Channel 1 Descriptor Control Register

Offset: 6C - 6Fh *Default Value:* 0000_0000h		*Size:* 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for DCR0 described in Section 13.8.1.12.				

13.8.1.25 Offset 80 - 83h: CCR2 – Channel 2 Channel Control Register

Table 456. Offset 80 - 83h: CCR2 – Channel 2 Channel Control Register

Offset: 80 - 83h *Default Value:* 0000_0000h		*Size:* 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for CCR0 described in Section 13.8.1.1				

13.8.1.26 Offset 84 - 87h: CSR2 – Channel 2 Channel Status Register

Table 457. Offset 84 - 87h: CSR2 – Channel 2 Channel Status Register

Offset: 84 -87h *Default Value:* 0000_0000h		*Size:* 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for CSR0 described in Section 13.8.1.2.				

13.8.1.27 Offset 88 - 8Bh: CDAR2 – Channel 2 Current Descriptor Address Register

Table 458. Offset 88 - 8Bh: CDAR2 – Channel 2 Current Descriptor Address Register

Offset: *88 - 8Bh* ***Size:*** *32 bit* ***Default Value:*** *0000_0000h*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for CDAR0 described in Section 13.8.1.3.				

13.8.1.28 Offset 8C - 8Fh: CDUAR2 – Channel 2 Current Descriptor Upper Address Register

Table 459. Offset 8C - 8Fh: CDUAR2 – Channel 2 Current Descriptor Upper Address Register

Offset: *8C - 8Fh* ***Size:*** *32 bit* ***Default Value:*** *0000_0000h*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for CDUAR0 described in Section 13.8.1.4.				

13.8.1.29 Offset 90 - 93h: SAR2 – Channel 2 Source Address Register

Table 460. Offset 90 - 93h: SAR2 – Channel 2 Source Address Register

Offset: *90 - 93h* ***Size:*** *32 bit* ***Default Value:*** *0000_0000h*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for SAR0 described in Section 13.8.1.5.				

13.8.1.30 Offset 94 - 97h: SUAR2 – Channel 2 Source Upper Address Register

Table 461. Offset 94 - 97h: SUAR2 – Channel 2 Source Upper Address Register

Offset: *94 - 97h* ***Size:*** *32 bit* ***Default Value:*** *0000_0000h*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for SUAR0 described in Section 13.8.1.6.				

13.8.1.31 Offset 98 - 9Bh: DAR2 – Channel 2 Destination Address Register

Table 462. Offset 98 - 9Bh: DAR2 – Channel 2 Destination Address Register

Offset: 98 - 9Bh		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for DAR0 described in Section 13.8.1.7.				

13.8.1.32 Offset 9C - 9Fh: DUAR2 – Channel 2 Destination Upper Address Register

Table 463. Offset 9C - 9Fh: DUAR2 – Channel 2 Destination Upper Address Register

Offset: 9C - 9Fh		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for DUAR0 described in Section 13.8.1.8.				

13.8.1.33 Offset A0 - A3h: NDAR2 – Channel 2 Next Descriptor Address Register

Table 464. Offset A0 - A3h: NDAR2 – Channel 2 Next Descriptor Address Register

Offset: A0 - A3h		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for NDAR0 described in Section 13.8.1.9.				

13.8.1.34 Offset A4 - A7h: NDUAR2 – Channel 2 Next Descriptor Upper Address Register

Table 465. Offset A4 - A7h: NDUAR2 – Channel 2 Next Descriptor Upper Address Register

Offset: A4 - A7h		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for NDUAR0 described in Section 13.8.1.10.				

13.8.1.35 Offset A8 - ABh: TCR2 – Channel 2 Transfer Count Register

Table 466. Offset A8 - ABh: TCR2 – Channel 2 Transfer Count Register

Offset: *A8 - ABh* ***Default Value:*** *0000_0000h*		***Size:*** *32 bit*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for TCR0 described in Section 13.8.1.11.				

13.8.1.36 Offset AC - AFh: DCR2 – Channel 2 Descriptor Control Register

Table 467. Offset AC - AFh: DCR2 – Channel 2 Descriptor Control Register

Offset: *AC - AFh* ***Default Value:*** *0000_0000h*		***Size:*** *32 bit*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for DCR0 described in Section 13.8.1.12.				

13.8.1.37 Offset C0 - C3h: CCR3 – Channel 3 Channel Control Register

Table 468. Offset C0 - C3h: CCR3 – Channel 3 Channel Control Register

Offset: *C0 - C3h* ***Default Value:*** *0000_0000h*		***Size:*** *32 bit*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for CCR0 described in Section 13.8.1.1.				

13.8.1.38 Offset C4 - C7h: CSR3 – Channel 3 Channel Status Register

Table 469. Offset C4 - C7h: CSR3 – Channel 3 Channel Status Register

Offset: *C4 - C7h* ***Default Value:*** *0000_0000h*		***Size:*** *32 bit*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for CSR0 described in Section 13.8.1.2.				

13.8.1.39 Offset C8 - CBh: CDAR3 – Channel 3 Current Descriptor Address Register

Table 470. Offset C8 - CBh: CDAR3 – Channel 3 Current Descriptor Address Register

Offset: C8 - CBh **Default Value:** 0000_0000h		**Size:** 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for CDAR0 described in Section 13.8.1.3.				

13.8.1.40 Offset CC - CFh: CDUAR3 – Channel 3 Current Descriptor Upper Address Register

Table 471. Offset CC - CFh: CDUAR3 – Channel 3 Current Descriptor Upper Address Register

Offset: CC - CFh **Default Value:** 0000_0000h		**Size:** 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for CDUAR0 described in Section 13.8.1.4.				

13.8.1.41 Offset D0 - D3h: SAR3 – Channel 3 Source Address Register

Table 472. Offset D0 - D3h: SAR3 – Channel 3 Source Address Register

Offset: D0 - D3h **Default Value:** 0000_0000h		**Size:** 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for SAR0 described in Section 13.8.1.5.				

13.8.1.42 Offset D4 - D7h: SUAR3 – Channel 3 Source Upper Address Register

Table 473. Offset D4 - D7h: SUAR3 – Channel 3 Source Upper Address Register

Offset: D4 - D7h **Default Value:** 0000_0000h		**Size:** 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for SUAR0 described in Section 13.8.1.6.				

13.8.1.43 Offset D8 - DBh: DAR3 – Channel 3 Destination Address Register

Table 474. Offset D8 - DBh: DAR3 – Channel 3 Destination Address Register

Offset: D8 - DBh		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for DAR0 described in Section 13.8.1.7.				

13.8.1.44 Offset DC - DFh: DUAR3 – Channel 3 Destination Upper Address Register

Table 475. Offset DC - DFh: DUAR3 – Channel 3 Destination Upper Address Register

Offset: DC - DFh		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for DUAR0 described in Section 13.8.1.8.				

13.8.1.45 Offset E0 - E3h: NDAR3 – Channel 3 Next Descriptor Address Register

Table 476. Offset E0 - E3h: NDAR3 – Channel 3 Next Descriptor Address Register

Offset: E0 - E3h		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for NDAR0 described in Section 13.8.1.9.				

13.8.1.46 Offset E4 - E7h: NDUAR3 – Channel 3 Next Descriptor Upper Address Register

Table 477. Offset E4 - E7h: NDUAR3 – Channel 3 Next Descriptor Upper Address Register

Offset: E4 - E7h		*Size:* 32 bit		
Default Value: 0000_0000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for NDUAR0 described in Section 13.8.1.10.				

13.8.1.47 Offset E8 - EBh: TCR3 – Channel 3 Transfer Count Register

Table 478. Offset E8 - EBh: TCR3 – Channel 3 Transfer Count Register

Offset: E8 - EBh *Default Value:* 0000_0000h		*Size:* 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for TCR0 described in Section 13.8.1.11.				

13.8.1.48 Offset EC - EFh: DCR3 – Channel 3 Descriptor Control Register

Table 479. Offset EC - EFh: DCR3 – Channel 3 Descriptor Control Register

Offset: EC - EFh *Default Value:* 0000_0000h		*Size:* 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
The bit descriptions for this register are identical to those for DCR0 described in Section 13.8.1.12.				

13.8.1.49 Offset 100 - 103h: DCGC – EDMA Controller Global Command

This register controls enabling and designation of priority channel.

Table 480. Offset 100 - 103h: DCGC – EDMA Controller Global Command

Offset: 100 - 103h *Default Value:* 0000_0000h		*Size:* 32 bit		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:03	Reserved	Reserved	0	
02	PCENBL	**Priority Channel Enable:** 0 = No priority channel. The Priority Channel Select bits are ignored. 1 = Enable the Priority Channel as programmed by the Priority Channel Select.	0b	RW
01:00	PCSLT	**Priority Channel Selects:** When Priority Channel Enable is set, the EDMA channel selected by this field has a higher priority than the others. 00 Channel 0 is the Priority Channel 01 Channel 1 is the Priority Channel 10 Channel 2 is the Priority Channel 11 Channel 3 is the Priority Channel	00b	RW

13.8.1.50 Offset 104 - 107h: DCGS – EDMA Controller Global Status

This register is accessed by the device driver to determine the source of an interrupt from the EDMA controller.

Table 481. Offset 104 - 107h: DCGS – EDMA Controller Global Status

Offset: *104 - 107h* ***Size:*** *32 bit*

Default Value: *0000_0000h*

Bits	Name	Description	Reset Value	Access
31:26	Reserved	Reserved	00h	
25	NIC3	**Normal Interrupt Condition from Channel 3:** 0 = No Channel 3 Normal Interrupt is generated. 1 = A Channel 3 Normal Interrupt has been generated.	0b	RO
24	EIC3	**Error interrupt Condition from Channel 3:** 0 = No Channel 3 Error Interrupt is generated. 1 = A Channel 3 Error Interrupt has been generated. The channel is in Abort status.	0b	RO
23:18	Reserved	Reserved	00h	
17	NIC2	**Normal Interrupt Condition from Channel 2:** 0 = No Channel 2 Normal Interrupt is generated. 1 = A Channel 2 Normal Interrupt has been generated.	0b	RO
16	EIC2	**Error interrupt Condition from Channel 2:** 0 = No Channel 2 Error Interrupt is generated. 1 = A Channel 2 Error Interrupt has been generated. The channel is in Abort status.	0b	RO
15:10	Reserved	Reserved	00h	
09	NIC1	**Normal Interrupt Condition from Channel 1:** 0 = No Channel 1 Normal Interrupt is generated. 1 = A Channel 1 Normal Interrupt has been generated.	0b	RO
08	EIC1	**Error interrupt Condition from Channel 1:** 0 = No Channel 1 Error Interrupt is generated. 1 = A Channel 1 Error Interrupt has been generated. The channel is in Abort status.	0b	RO
07:02	Reserved	Reserved	00h	
01	NIC0	**Normal Interrupt Condition from Channel 0:** 0 = No Channel 0 Normal Interrupt is generated. 1 = A Channel 0 Normal Interrupt has been generated.	0b	RO
00	EIC0	**Error interrupt Condition from Channel 0:** 0 = No Channel 0 Error Interrupt is generated. 1 = A Channel 0 Error Interrupt has been generated. The channel is in Abort status.	0b	RO

14.0 Bridging and Configuration (IICH)

14.1 Configuration Registers (Memory Space)

This section describes all registers and base functionality that are related to configuration and not a specific interface (such as LPC, PCI, or PCI Express). It contains the root complex register block, which describes the behavior of the upstream internal link.

This block is mapped into memory space using register RCBA of the PCI to LPC bridge. Accesses in this space must be limited to 32-bit (Dword) quantities. Burst accesses are not allowed.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 482. Bridging and Configuration Register (Memory Space) Summary Table (Sheet 1 of 2)

Address		Symbol	Register Name/Function	Default	Access
Start	End				
VC Configuration Registers					
0000h	0003Fh	VCH	Virtual Channel Capability Header Register	10010002h	RO
0004h	0007h	VCAP1	Virtual Channel Capability 1 Register	0801h	RO
0008h	000Bh	VCAP2	Virtual Channel Capability 2 Register	0001h	RO
000Ch	000Dh	PVC	Port Virtual Channel Control Register	0	RO, RW
000Eh	000Fh	PVS	Port Virtual Channel Status Register	0	RO
0010h	0013h	V0CAP	Virtual Channel 0 Resource Capability Register	0001h	RO
0014h	0017h	V0CTL	Virtual Channel 0 Resource Control Register	800000FFh	RO, RW
001Ah	001Bh	V0STS	Virtual Channel 0 Resource Status Register	0	RO
Root Complex Topology Configuration Registers					
0100h	0103h	RCTCL	Root Complex Topology Capability List Register	1A010005h	RO
0104h	0107h	ESD	Element Self Description Register	00000602h	RO, RWO
0110h	0113h	ULD	Upstream Link Descriptor Register	0001h	RO, RWO
0118h	011Fh	ULBA	Upstream Link Base Address Register	00000000_00000000h	RWO
0120h	0123h	RPB0D	Root PortB0 (PEB0) Descriptor Register	See desc	RO
0128h	012Fh	RPB0BA	Root PortB0 (PEB0) Base Address Register	00000000_00 0E0000h	RO
0130h	0133h	RPB1D	Root PortB1 (PEB1) Descriptor Register	See desc	RO
0138h	013Fh	RPB1BA	Root PortB1 (PEB1) Base Address Register	00000000_00 0E1000h	RO
0140h	0143h	RPB2D	Root Port B2 (PEB2) Descriptor Register	See desc	RO

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 482. Bridging and Configuration Register (Memory Space) Summary Table (Sheet 2 of 2)

Address		Symbol	Register Name/Function	Default	Access
Start	End				
0148h	014Fh	RPB2BA	Root Port B2 (PEB2) Base Address Register	00000000_000E2000h	RO
0150h	0153h	RPB3D	Root Port B3 (PEB3) Descriptor Register	See desc	RO
0158h	015Fh	RPB3BA	Root Port B3 (PEB3) Base Address Register	00000000_000E3000h	RO
Internal Link Configuration Registers					
01A0h	01A3h	ILCL	Internal Link Capability List Register	00010006h	RO
01A4h	01A7h	LCAP	Link Capabilities Register	0012441h	RO
01A8h	01A9h	LCTL	Link Control Register	0h	RO, RW
01AAh	01ABh	LSTS	Link Status Register	0041h	RO
I/O Data Bus Configuration Registers					
0224h	0227h	RPC	Root Port Configuration Register	00000000h	RO, RW
TCO Configuration Register					
3000h	3001Fh	TCTL	TCO Control Register	0h	RO, RW
Interrupt Configuration Registers					
3100h	3103h	D31IP	Device 31 Interrupt Pin Register	0042210h	RO, RW
3104h	3107h	D30IP	Device 30 Interrupt Pin Register	00002100h	RO
3108h	310Bh	D29IP	Device 29 Interrupt Pin Register	10004321h	RO, RW
310Ch	310Fh	D28IP	Device 28 Interrupt Pin Register	00004321h	RO, RW
3140h	3141h	D31IR	Device 31 Interrupt Route Register	03210h	RO, RW
3142h	3143h	D30IR	Device 30 Interrupt Route Register	03210h	RO, RW
3144h	3145h	D29IR	Device 29 Interrupt Route Register	03210h	RO, RW
3146h	3147h	D28IR	Device 28 Interrupt Route Register	03210h	RO, RW
31FFh	31FFh	OIC	Other Interrupt Control Register	0	RO, RW
General Configuration Registers					
3400h	3403h	RC	RTC Configuration Register	0	RO, RW, RWO
3404h	3407h	HPTC	High Performance Precision Timer Configuration Register	0	RO, RW
3410h	3413h	GCS	General Control and Status Register	See desc	RO, WO, RW
3414h	3417h	BUC	Backed-Up Control Register	See desc	RO, RW
3418hh	341Bh	FD	Function Disable Register	00000C12h	RO, RW
341C	341Fh	PRC	Power Reduction Control Register Clock Gating	0	RO, RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

14.1.1 VC Configuration Registers

14.1.1.1 Offset 0000 - 0003h: VCH – Virtual Channel Capability Header Register

Table 483. Offset 0000 - 0003h: VCH – Virtual Channel Capability Header Register

Offset: 0000 - 0003Fh ***Size:*** *32 bit*

Default Value: *10010002h* ***Power Well:***

Bits	Name	Description	Reset Value	Access
31:20	NCO	**Next Capability Offset:** Indicates the next item in the list.	100h	RO
19:16	CV	**Capability Version:** Indicates this is version 1 of the capability structure by the PCI SIG.	1h	RO
15:00	CID	**Capability ID:** Indicates this is the Virtual Channel capability item.	0002h	RO

14.1.1.2 Offset 0004 - 0007h: VCAP1 – Virtual Channel Capability 1 Register

Table 484. Offset 0004 - 0007h: VCAP1 – Virtual Channel Capability 1 Register

Offset: 0004 - 0007h ***Size:*** *32 bit*

Default Value: *0801h* ***Power Well:***

Bits	Name	Description	Reset Value	Access
31:12	Reserved	Reserved	0	
11:10	Reserved	Reserved	10b	
09:08	RC	**Reference Clock:** Fixed at 100 ns for this version of the *PCI Express Base Specification*.	00b	RO
07	Reserved	Reserved	0	
06:04	LPEVC	**Low Priority Extended VC Count:** Indicates that there are no additional VCs of low priority with extended capabilities.	000	RO
03	Reserved	Reserved	0	
02:00	EVC	Reserved.	001	

14.1.1.3 Offset 0008 - 000Bh: VCAP2 – Virtual Channel Capability 2 Register

Table 485. Offset 0008 - 000Bh: VCAP2 – Virtual Channel Capability 2 Register

Offset: 0008 - 000Bh ***Size:*** *32 bit*

Default Value: *0001h* ***Power Well:***

Bits	Name	Description	Reset Value	Access
31:24	ATO	**VC Arbitration Table Offset:** Indicates that no table is present for VC arbitration since it is fixed.	00h	RO
23:08	Reserved	Reserved	0	
07:00	AC	**VC Arbitration Capability:** Indicates that the VC arbitration is fixed in the root complex.	01h	RO

14.1.1.4 Offset 000C - 000Dh: PVC – Port Virtual Channel Control Register

Table 486. Offset 000C - 000Dh: PVC – Port Virtual Channel Control Register

Offset: 000C - 000Dh			***Size:*** *16 bit*	
Default Value: *0*			***Power Well:***	
Bits	**Name**	**Description**	**Reset Value**	**Access**
15:04	Reserved	Reserved	0	
03:01	AS	**VC Arbitration Select:** Indicates which VC must be programmed in the VC arbitration table. The root complex takes no action on the setting of this field since there is no arbitration table.	000	RW
00	LAT	**Load VC Arbitration Table:** Indicates that the table programmed must be loaded into the VC arbitration table. This bit is defined as read/write with always returning 0 on reads. Since there is no VC arbitration table in the root complex, this bit can be built as read-only.	0	RO

14.1.1.5 Offset 000E - 000Fh: PVS – Port Virtual Channel Status Register

Table 487. Offset 000E - 000Fh: PVS – Port Virtual Channel Status Register

Offset: 000E - 000Fh			***Size:*** *16 bit*	
Default Value: *0*			***Power Well:***	
Bits	**Name**	**Description**	**Reset Value**	**Access**
15:01	Reserved	Reserved	0	
00	VAS	**VC Arbitration Table Status:** Indicates the coherency status of the VC Arbitration table when it is being updated. This field is hardwired to 0 in the root complex since there is no VC arbitration table.	0	RO

14.1.1.6 Offset 0010 - 00013h: V0CAP – Virtual Channel 0 Resource Capability Register

Table 488. Offset 0010 - 00013h: V0CAP – Virtual Channel 0 Resource Capability Register

Offset: 0010 - 00013h			***Size:*** *32 bit*	
Default Value: 0001h			***Power Well:***	
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:24	AT	**Port Arbitration Table Offset:** This VC implements no port arbitration table since the arbitration is fixed.	0	RO
23	Reserved	Reserved	0	
22:16	MTS	**Maximum Time Slots:** This VC implements fixed arbitration, and therefore this field is not used.	00h	RO
15	RTS	**Reject Snoop Transactions:** This VC must be able to take snoopable transactions.	0	RO
14	APS	**Advanced Packet Switching:** This VC is capable of all transactions, not just advanced packet switching transactions.	0	RO
13:08	Reserved	Reserved	0	
07:00	PAC	**Port Arbitration Capability:** Indicates that this VC uses fixed port arbitration.	01h	RO

14.1.1.7 Offset 0014 - 00017h: V0CTL – Virtual Channel 0 Resource Control Register

Table 489. Offset 0014 - 00017h: V0CTL – Virtual Channel 0 Resource Control Register

Offset: 0014 - 00017h			***Size:*** *32 bit*	
Default Value: 800000FFh			***Power Well:***	
Bits	**Name**	**Description**	**Reset Value**	**Access**
31	EN	**Virtual Channel Enable:** Enables the VC when set. Disables the VC when cleared. 0 = Disables the VC 1 = Enables the VC	1	RO
30:27	Reserved	Reserved	0	
26:24	ID	**Virtual Channel Identifier:** Indicates the ID to use for this virtual channel.	000	RO
23:20	Reserved	Reserved	0	
19:17	PAS	**Port Arbitration Select:** Indicates which port table is being programmed. The root complex takes no action on this setting since the arbitration is fixed and there is no arbitration table.	0	RW
16	LAT	**Load Port Arbitration Table:** The root complex does not implement an arbitration table for this virtual channel.	0	RO
15:08	Reserved	Reserved	0	
07:01	TCVC0M	**Transaction Class / Virtual Channel 0 Map:** Indicates which transaction classes are mapped to this virtual channel. 0 = This transaction class is not mapped to the virtual channel 0. 1 = This transaction class is mapped to the virtual channel 0.	7Fh	RW
00	TC0VC0M	**Transaction Class 0 / Virtual Channel 0 Map:** Indicates that transaction class 0 is always mapped to VC0.	1	RO

14.1.1.8 Offset 001A - 001Bh: V0STS – Virtual Channel 0 Resource Status Register

Table 490. Offset 001A - 001Bh: V0STS – Virtual Channel 0 Resource Status Register

Offset: 001A - 001Bh			***Size:*** *16 bit*	
Default Value: *0*			***Power Well:***	
Bits	**Name**	**Description**	**Reset Value**	**Access**
15:02	Reserved	Reserved	0	
01	NP	**VC Negotiation Pending:** 0 = Indicates the virtual channel is not being negotiated with ingress ports. 1 = Indicates the virtual channel is still being negotiated with ingress ports.	0	RO
00	ATS	**Port Arbitration Table Status:** There is no port arbitration table for this VC so this bit is reserved at 0.	0	RO

This is a 64-byte register that contains the arbitration table to be loaded into the port arbitration table. Every four bits contains an entry for one of the downstream PCI Express ports or a 0h to indicate idle. The ports are mapped as follows:

- Port: Value used is 1h
- Port: Value used is 2h
- Port: Value used is 3h
- Port: Value used is 4h

This table is copied to an internal structure used during port arbitration when V1CTL.PAS is set to 04h, and V1CTL.LAT is set to '1'.

14.1.2 Root Complex Topology Capability Structure Registers

The following registers follow the PCI Express capability list structure as defined in the *PCI Express Base Specification*, to indicate the capabilities of NSI.

14.1.2.1 Offset 0100 - 0103h: RCTCL – Root Complex Topology Capabilities List Register

Table 491. Offset 0100 - 0103h: RCTCL – Root Complex Topology Capabilities List Register

Offset: 0100 - 0103h	***Size:*** *32 bit*
Default Value: 1A010005h	***Power Well:***

Bits	Name	Description	Reset Value	Access
31:20	NEXT	**Next Capability:** Indicates next item in the list.	1A0h	RO
19:16	CV	**Capability Version:** Indicates the version of the capability structure.	1h	RO
15:00	CID	**Capability ID:** Indicates this is a PCI Express link capability section of an RCRB.	0005h	RO

14.1.2.2 Offset 0104 - 0107h: ESD – Element Self Description Register

Table 492. Offset 0104 - 0107h: ESD – Element Self Description Register

Offset: 0104 - 0107h	***Size:*** *32 bit*
Default Value: 00000602h	***Power Well:***

Bits	Name	Description	Reset Value	Access
31:24	PN	**Port Number:** A value of 0 to indicate the egress port for the IICH.	00h	RO
23:16	CID	**Component ID:** Indicates the component ID assigned to this element by software. This is written once by platform BIOS and is locked until a platform reset.	00h	RWO
15:08	NLE	**Number of Link Entries:** Indicates that one link entry, corresponding to NSI, and four root port entries, for the downstream port, and one reserved entry are described by this RCRB.	06h	RO
07:04	Reserved	Reserved	0h	
03:00	ET	**Element Type:** Indicates that the element type is a root complex internal link.	2h	RO

14.1.2.3 Offset 0110 - 0113h: ULD – Upstream Link Description Register

Table 493. Offset 0110 - 0113h: ULD – Upstream Link Description Register

Offset: 0110 - 0113h	***Size:*** *32 bit*
Default Value: 0001h	***Power Well:***

Bits	Name	Description	Reset Value	Access
31:24	PN	**Target Port Number:** This field is programmed by platform BIOS to match the port number of the IMCH. RCRB that is attached to this RCRB.	00h	RWO
23:16	TCID	**Target Component ID:** This field is programmed by platform BIOS to match the component ID of the IMCH. RCRB that is attached to this RCRB.	00h	RWO
15:02	Reserved	Reserved	0	
01	LT	**Link Type:** Indicates that the link points to the IMCH RCRB.	0	RO
00	LV	**Link Valid:** Indicates that this link entry is valid.	1	RO

14.1.2.4 Offset 0118 - 011Fh: ULBA - Upstream Link Base Address Register

Table 494. Offset 0118 - 011Fh: ULBA - Upstream Link Base Address Register

Offset: 0118 - 011Fh	***Size:*** *64 bit*
Default Value: *00000000_00000000h*	***Power Well:***

Bits	Name	Description	Reset Value	Access
63:32	BAU	**Base Address Upper:** This field is programmed by platform BIOS to match the upper 32-bits of base address of the IMCH. RCRB that is attached to this RCRB.	00000000h	RWO
31:00	BAL	**Base Address Lower:** This field is programmed by platform BIOS to match the lower 32-bits of base address of the IMCH. RCRB that is attached to this RCRB.	00000000h	RWO

14.1.2.5 Offset 0120 - 0123h: RPB0D – Root Port B0 (PEB0) Description Register

Table 495. Offset 0120 - 0123h: RPB0D – Root Port B0 (PEB0) Description Register

Offset: 0120 - 0123h	***Size:*** *32 bit*
Default Value: See bit descriptions below	***Power Well:***

Bits	Name	Description	Reset Value	Access
31:24	PN	**Target Port Number:** Indicates the target port number is 1h (root port B0).	01h	RO
23:16	TCID	**Target Component ID:** This field returns the value of the ESD.CID field programmed by platform BIOS, since the root port is in the same component as the RCRB.	Init	RO
15:02	Reserved	Reserved	0	
01	LT	**Link Type:** Indicates that the link points to a root port.	1	RO
00	LV	**Link Valid:** When FD.PE1D is set, this link is not valid (returns 0). When FD.PE1D is cleared, this link is valid (returns 1).	0/1	RO

14.1.2.6 Offset 0128 - 012Fh: RPB0BA – Root Port B0 (PEB0) Base Address Register

Table 496. Offset 0128 - 012Fh: RPB0BA – Root Port B0 (PEB0) Base Address Register

Offset: 0128 - 012Fh	***Size:*** *64 bit*
Default Value: 00000000_000E0000h	***Power Well:***

Bits	Name	Description	Reset Value	Access
63:32	CBAU	**Config Space Base Address Upper:** This field is reserved in IICH implementations.	00000000h	RO
31:28	CBAL	**Config Space Base Address Lower:** This field is reserved in IICH implementations.	0h	RO
27:20	BN	**Bus Number:** Indicates the root port is on bus 0.	00h	RO
19:15	DN	**Device Number:** Indicates the root port is on device 28.	1Ch	RO
14:12	FN	**Function Number:** Indicates the root port is on function 0.	0h	RO
11:00	Reserved	Reserved	000h	

14.1.2.7 Offset 0130 - 0133h: RPB1D – Root Port B1(PEB1) Description Register

Table 497. Offset 0130 - 0133h: RPB1D – Root Port B1(PEB1) Description Register

Offset: 0130 - 0133h	***Size:*** *32 bit*
Default Value: See bit descriptions below	***Power Well:***

Bits	Name	Description	Reset Value	Access
31:24	PN	**Target Port Number:** Indicates the target port number is 2h (root port B1).	02h	RO
23:16	TCID	**Target Component ID:** This field returns the value of the ESD.CID field programmed by platform BIOS since the root port is in the same component as the RCRB.	Init	RO
15:02	Reserved	Reserved	0	
01	LT	**Link Type:** 0 = Indicates that the link does not point to a root port. 1 = Indicates that the link points to a root port	1	RO
00	LV	**Link Valid:** When RPC.PC is ‘01’, ‘10’, or ‘11’, or FD.PE2D is set, the link for this root port is not valid (return ‘0’). When RPC.PC is ‘00’ and FD.PE2D is cleared, the link for this root port is valid (return ‘1’).	0/1	RO

14.1.2.8 Offset 0138 - 013Fh: RPB1BA – Root Port B1 (PEB1) Base Address Register

Table 498. Offset 0138 - 013Fh: RPB1BA – Root Port B1 (PEB1) Base Address Register

Offset: 0138 - 013Fh	***Size:*** *64 bit*
Default Value: 00000000_000E1000h	***Power Well:***

Bits	Name	Description	Reset Value	Access
63:32	CBAU	**Config Space Base Address Upper:** This field is reserved in IICH implementations.	00000000h	
31:28	CBAL	**Config Space Base Address Lower:** This field is reserved in IICH implementations.	0h	
27:20	BN	**Bus Number:** Indicates the root port is on bus 0.	00h	RO
19:15	DN	**Device Number:** Indicates the root port is on device 28.	1Ch	RO
14:12	FN	**Function Number:** Indicates the root port is on function 1.	1h	RO
11:00	Reserved	Reserved	000h	

14.1.2.9 Offset 0140 - 0143h: RPB2D – Root Port B2 (PEB2) Description Register

Table 499. Offset 0140 - 0143h: RPB2D – Root Port B2 (PEB2) Description Register

Offset: 0140 - 0143h	***Size:*** *32 bit*
Default Value: See bit descriptions below	***Power Well:***

Bits	Name	Description	Reset Value	Access
31:24	PN	**Target Port Number:** Indicates the target port number is 3h (root port B2).	03h	RO
23:16	TCID	**Target Component ID:** This field returns the value of the ESD. CID field programmed by platform BIOS, since the root port is in the same component as the RCRB.	Init	RO
15:02	Reserved	Reserved	0	
01	LT	**Link Type:** 0 = Indicates that the link does not point to a root port. 1 = Indicates that the link points to a root port.	1	RO
00	LV	**Link Valid:** When RPC.PC is '11', or.FD.PE3D is set, the link for this root port is not valid (return '0'). When RPC.PC is '00', '01', or '10', and FD.PE3D is cleared, the link for this root port is valid (return '1').	0/1	RO

14.1.2.10 Offset 0148 - 014Fh: RPB2BA – Root Port B2 (PEB2) Base Address Register1

Table 500. Offset 0148 - 014Fh: RPB2BA – Root Port B2 (PEB2) Base Address Register1

Offset: 0148 - 014Fh			***Size:*** *64 bit*	
Default Value: 00000000_000E2000h			***Power Well:***	
Bits	**Name**	**Description**	**Reset Value**	**Access**
63:32	CBAU	**Config Space Base Address Upper:** This field is reserved in IICH implementations.	00000000h	
31:28	CBAL	**Config Space Base Address Lower:** This field is reserved in IICH implementations.	0h	
27:20	BN	**Bus Number:** Indicates the root port is on bus 0.	00h	RO
19:15	DN	**Device Number:** Indicates the root port is on device 28.	1Ch	RO
14:12	FN	**Function Number:** Indicates the root port is on function 2.	2h	RO
11:00	Reserved	Reserved	000h	

14.1.2.11 Offset 0150 - 0153h: RPB3D – Root Port B3 (PEB3) Description Register

Table 501. Offset 0150 - 0153h: RPB3D – Root Port B3 (PEB3) Description Register

Offset: 0150 - 0153h			***Size:*** *32 bit*	
Default Value: See bit descriptions below			***Power Well:***	
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:24	PN	**Target Port Number:** Indicates the target port number is 4h (root port B3).	04h	RO
23:16	TCID	**Target Component ID:** This field returns the value of the ESD.CID field programmed by platform BIOS, since the root port is in the same component as the RCRB.	Init	RO
15:02	Reserved	Reserved	0	
01	LT	**Link Type:** 0 = Indicates that the link does not point to a root port. 1 = Indicates that the link points to a root port.	1	RO
00	LV	**Link Valid:** When RPC.PC is ‘10’ or ‘11’, or FD.PE4D is set, the link for this root port is not valid (return ‘0’). When RPC.PC is ‘00’ or ‘01’, and FE.PE4D is cleared, the link for this root port is valid (return ‘1’).	0/1	RO

14.1.2.12 Offset 0158 - 015Fh: RPB3BA – Root Port B3 (PEB3) Base Address Register

Table 502. Offset 0158 - 015Fh: RPB3BA – Root Port B3 (PEB3) Base Address Register

Offset: 0158 - 015Fh	***Size:*** *64 bit*
Default Value: *00000000_000E3000h*	***Power Well:***

Bits	Name	Description	Reset Value	Access
63:32	CBAU	**Config Space Base Address Upper:** This field is reserved in IICH implementations.	00000000h	
31:28	CBAL	**Config Space Base Address Lower:** This field is reserved in IICH implementations.	0h	
27:20	BN	**Bus Number:** Indicates the root port is on bus 0.	00h	RO
19:15	DN	**Device Number:** Indicates the root port is on device 28.	1Ch	RO
14:12	FN	**Function Number:** Indicates the root port is on function 3.	3h	RO
11:00	Reserved	Reserved	000	

14.1.3 Internal Link Configuration Registers

14.1.3.1 Offset 01A0 - 01A3h: ILCL – Internal Link Capabilities List Register

Table 503. Offset 01A0 - 01A3h: ILCL – Internal Link Capabilities List Register

Offset: 01A0 - 01A3h	***Size:*** *32 bit*
Default Value: *00010006h*	***Power Well:***

Bits	Name	Description	Reset Value	Access
31:20	NEXT	**Next Capability:** Indicates this is the last item in the list.	000h	RO
19:16	CV	**Capability Version:** Indicates the version of the capability structure.	1h	RO
15:00	CID	**Capability ID:** Indicates this is the capability for NSI.	0006h	RO

14.1.3.2 Offset 01A4 - 01A7h: LCAP – Link Capabilities Register

Table 504. Offset 01A4 - 01A7h: LCAP – Link Capabilities Register (Sheet 1 of 2)

Offset: 01A4 - 01A7h	***Size:*** *32 bit*
Default Value: 0012441h	***Power Well:***

Bits	Name	Description	Reset Value	Access
31:18	Reserved	Reserved	0	
17:15	EL1	**Intel® 3100 Chipset does not support L0s or L1.** **L1 Exit Latency:** Indicates that the exit latency is 2 µs to 4 µs.	010	RO
14:12	EL0	**Intel® 3100 Chipset does not support L0s or L1.** **L0s Exit Latency:** This field is read/write and updatable by BIOS. It defaults to 128 ns to less than 256 ns, assuming a common-clock configuration between IICH and IMCH. If a unique clock value is used, it is recommended that BIOS update this field to 100 (512 ns to less than 1 µs). When BIOS set sets this field, it must also update NSI's DBG.NFTS field, located at offset 2024h in IICH configuration space.	010	RW

Table 504. Offset 01A4 - 01A7h: LCAP – Link Capabilities Register (Sheet 2 of 2)

Offset: 01A4 - 01A7h		**Size:** 32 bit		
Default Value: 0012441h		**Power Well:**		
Bits	**Name**	**Description**	**Reset Value**	**Access**
11:10	APMS	**Intel® 3100 Chipset does not support L0s or L1.** **Active State Link PM Support:** Indicates the level of active state power management on NSI. Bits Definition 00 Neither L0s nor L1 supported 01 L0s Entry supported (Per PCI Express spec, L0s must be supported, but the Intel® 3100 Chipset has defeatured L0s.) 10 Reserved: L1 Entry not supported on NSI 11 Reserved: L1 Entry not supported on NSI	1h	RWO
09:04	MLW	**Maximum Link Width:** Indicates the maximum link width is four ports.	4h	RO
03:00	MLS	**Maximum Link Speed:** Indicates the link speed is 2.5 Gbits/s.	1h	RO

14.1.3.3 Offset 01A8 - 01A9h: LCTL – Link Control Register

Table 505. Offset 01A8 - 01A9h: LCTL – Link Control Register

Offset: 01A8 - 01A9h		**Size:** 16 bit		
Default Value: 0h		**Power Well:**		
Bits	**Name**	**Description**	**Reset Value**	**Access**
15:07	Reserved	Reserved	0	
06:02	Reserved	Reserved	0	
01:00	APMC	L0s has been defeatured on Intel® 3100 Chipset, and ASPM must never be turned on. **Active State Link PM Control:** Indicates whether NSI must enter L1. Bits Definition 00 Disabled 01 L0s Entry Enabled 10 L1 Entry Enabled 11 L0s and L1 Entry Enabled	0h	RW

14.1.3.4 Offset 01AA - 01ABh: LSTS – Link Status Register

Table 506. Offset 01AA - 01ABh: LSTS – Link Status Register

Offset: 01AA - 01ABh		**Size:** 16 bit		
Default Value: 0041h		**Power Well:**		
Bits	**Name**	**Description**	**Reset Value**	**Access**
15:10	Reserved	Reserved	0	
09:04	NLW	**Negotiated Link Width:** Minimum negotiated link width is a x4 port. The contents of this register are undefined if the link has not successfully trained.	4h	RO
03:00	LS	**Link Speed:** Link is 2.5 Gbits/s.	1h	RO

14.1.4 I/O Data Bus Configuration Registers

Intel® 3100 Chipset

14.1.4.1 Offset 0224 - 0227h: RPC – Root Port Configuration Register

Table 507. Offset 0224 - 0227h: RPC – Root Port Configuration Register

Offset: 0224 - 0227h		***Size:*** *32 bit*		
Default Value: *00h*		***Power Well:***		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:08	Reserved	Reserved	000000h	
07	HPE	**High Priority Port Enable:** 0 = The high priority path is not enabled 1 = The port selected by the HPP field in this register is enabled for high priority.	0	RW
06	Reserved	Reserved	0	
05:04	HPP	**High Priority Port:** This controls which port is enabled for high priority when the HPE bit in this register is set. Bits Port 11 Port B3 10 Port B2 01 Port B1 00 Port B0	00	RW
03:02	Reserved	Reserved	00	
01:00	PC	**Port Configuration:** This controls how the PCI bridges are organized in various modes of operation, and represent the value of strap IICH_RPC[1]and IICH_RPC[0] when DFXTEST# is deasserted. Bit 1 represents the value of IICH_RPC[1]and bit 0 represents the value of IICH_RPC[0]These pins are sampled on the rising edge of PWROK. For the following mappings, if a port is not shown, it is considered a x1 port with no connection. Bits Mode Ports Routing 11 1 x4 Port B0 (x4) 10 Reserved 01 Reserved 00 4 x1s Port B0 (x1), Port B1 (x1), Port B2 (x1), Port B3 (x1) These bits live in the resume well and are only reset by RSMRST#.	00	RW

14.1.5 TCO Configuration

14.1.5.1 Offset 3000h: TCTL – TCO Control Register

Table 508. Offset 3000h: TCTL – TCO Control Register

Offset: 3000h		*Size:* 8 bit		
Default Value: 0h		*Power Well:*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
07	IE	**TCO IRQ Enable:** 0 = TCO IRQ is disabled. 1 = TCO IRQ is enabled, as selected by the TCO_IRQ_SEL field.	0	RW
06:03	Reserved	Reserved	0	
02:00	IS	**TCO IRQ Select:** Specifies on which IRQ the TCO internally appears. If not using the APIC, the TCO interrupt must be routed to IRQ9-11, and that interrupt is not sharable with the SERIRQ stream, but it can be shared with other PCI interrupts. If using the APIC, the TCO interrupt can also be mapped to IRQ20-23 and can be shared with other interrupt. Bits SCI Map 000 IRQ9 001 IRQ10 010 IRQ11 011 Reserved 100 IRQ20 (only if APIC enabled) 101 IRQ21 (only if APIC enabled) 110 IRQ22 (only if APIC enabled) 111 IRQ23 (only if APIC enabled) When setting these bits, the IE bit must be cleared to prevent glitches. When the interrupt is mapped to APIC interrupts 9, 10, or 11, the APIC must be programmed for active-high reception. When the interrupt is mapped to APIC interrupts 20 through 23, the APIC must be programmed for active-low reception.	000	RW

14.1.6 Interrupt Configuration Registers

14.1.6.1 Offset 3100 - 3103h: D31IP – Device 31 Interrupt Pin Register

Table 509. Offset 3100 - 3103h: D31IP – Device 31 Interrupt Pin Register

Offset: 3100 - 3103h		*Size:* 32 bit		
Default Value: 00042210h		*Power Well:*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:16	Reserved	Reserved	0	
15:12	SMIP	**SM Bus Pin:** See the CIP description. This field applies to the SM Bus controller.	2h	RW
11:08	SIP	**SATA Pin:** See the CIP description. This field applies to the SATA controller.	2h	RW
07:04	Reserved	Reserved	1h	
03:00	PIP	**PCI Bridge Pin:** See the CIP description. This field applies to the PCI bridge. Currently, the PCI bridge does not generate an interrupt so this field is read-only and ‘0’.	0h	RO

14.1.6.2 Offset 3104 - 3107h: D30IP – Device 30 Interrupt Pin Register

Table 510. Offset 3104 - 3107h: D30IP – Device 30 Interrupt Pin Register

Offset: 3104 - 3107h			***Size:*** *32 bit*	
Default Value: *00002100h*			***Power Well:***	
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:16	Reserved	Reserved	0	
15:12	Reserved	Reserved	2h	
11:08	Reserved	Reserved	1h	
07:04	Reserved	Reserved	0	
03:00	LIP	**LPC Bridge Pin:** See the AMIP description. This field applies to the LPC bridge. Currently, the LPC bridge does not generate an interrupt so this field is read-only and '0'.	0h	RO

14.1.6.3 Offset 3108 - 310Bh: D29IP – Device 29 Interrupt Pin Register

Table 511. Offset 3108 - 310Bh: D29IP – Device 29 Interrupt Pin Register

Offset: 3108 - 310Bh			***Size:*** *32 bit*	
Default Value: 10004321h			***Power Well:***	
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:28	EIP	**EHCI Pin:** Indicates which pin the EHCI controller drives as its interrupt. **Bits Pin Bits Pin** 0h No Interrupt 1h INTA# 2h INTB# 3h INTC# 4h INTD# 5h–Fh Reserved	1h	RW
27:16	Reserved	Reserved	0	
15:12	Reserved	Reserved	4h	
11:08	Reserved	Reserved	3h	
07:04	U1P	**UHCI 1 Pin:** See the EIP description. Applies to UCHI controller 1 (ports 2 and 3).	2h	RW
03:00	U0P	**UHCI 0 Pin:** See the EIP description. Applies to UCHI controller 0 (ports 0 and 1).	1h	RW

14.1.6.4 Offset 310C - 310Fh: D28IP – Device 28 Interrupt Pin Register

Table 512. Offset 310C - 310Fh: D28IP – Device 28 Interrupt Pin Register

Offset: 310C - 310Fh ***Size:*** *32 bit*

Default Value: 00004321h ***Power Well:***

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0	
15:12	P4IP	**PCI Express 4 Pin:** Indicates which pin PCI Express port B3 drives as its interrupt. Bits Pin Bits Pin 0h No Interrupt 1h INTA# 2h INTB# 3h INTC# 4h INTD# 5h–Fh Reserved	4h	RW
11:08	P3IP	**PCI Express 3 Pin:** Indicates which pin PCI Express port B2 drives as its interrupt.	3h	RW
07:04	P2IP	**PCI Express 2 Pin:** Indicates which pin PCI Express port B1 drives as its interrupt.	2h	RW
03:00	P1IP	**PCI Express 1 Pin:** Indicates which pin PCI Express port B0 drives as its interrupt.	1h	RW

14.1.6.5 Offset 3140 - 3141h: D31IR – Device 31 Interrupt Route Register

Table 513. Offset 3140 - 3141h: D31IR – Device 31 Interrupt Route Register

Offset: 3140 - 3141h ***Size:*** *16 bit*

Default Value: 03210h ***Power Well:***

Bits	Name	Description	Reset Value	Access
15	Reserved	Reserved	0	
14:12	IDR	**Interrupt D Pin Route:** Indicates which physical pin on the IICH is connected to the INTD# pin reported for device 31 functions. **Bits Pin Bits Pin** 0h PIRQA# 4h PIRQD# 1h PIRQB# 5h PIRQF# 2h PIRQC# 6h PIRQG# 3h PIRQD# 7h PIRQH#	3h	RW
11	Reserved	Reserved	0	
10:08	ICR	**Interrupt C Pin Route:** See the IDR description. This field applies to INTC#.	2h	RW
07	Reserved	Reserved	0	
06:04	IBR	**Interrupt B Pin Route:** See the IDR description. This field applies to INTB#.	1h	RW
03	Reserved	Reserved	0	
02:00	IAR	**Interrupt A Pin Route:** See the IDR description. This field applies to INTA#.	0h	RW

14.1.6.6 Offset 3142 - 3143h: D30IR – Device 30 Interrupt Route Register

Table 514. Offset 3142 - 3143h: D30IR – Device 30 Interrupt Route Register

Offset: 3142 - 3143h ***Size:*** *16 bit*

Default Value: 03210h ***Power Well:***

Bits	Name	Description	Reset Value	Access
15	Reserved	Reserved	0	
14:12	IDR	**Interrupt D Pin Route:** Indicates which physical pin on the IICH is connected to the INTD# pin reported for device 30 functions. **Bits Pin Bits Pin** 0h PIRQA# 4h PIRQD# 1h PIRQB# 5h PIRQF# 2h PIRQC# 6h PIRQG# 3h PIRQD# 7h PIRQH#	3h	RW
11	Reserved	Reserved	0	
10:08	ICR	**Interrupt C Pin Route:** See the IDR description. This field applies to INTC#.	2h	RW
07	Reserved	Reserved	0	
06:04	IBR	**Interrupt B Pin Route:** See the IDR description. This field applies to INTB#.	1h	RW
03	Reserved	Reserved	0	
02:00	IAR	**Interrupt A Pin Route:** See the IDR description. This field applies to INTA#.	0h	RO

14.1.6.7 Offset 3144 - 3145h: D29IR – Device 29 Interrupt Route Register

Table 515. Offset 3144 - 3145h: D29IR – Device 29 Interrupt Route Register

Offset: 3144 - 3145h ***Size:*** *16 bit*

Default Value: 03210h ***Power Well:***

Bits	Name	Description	Reset Value	Access
15	Reserved	Reserved	0	
14:12	IDR	**Interrupt D Pin Route:** Indicates which physical pin on the IICH is connected to the INTD# pin reported for device 29 functions. **Bits Pin Bits Pin** 0h PIRQA# 4h PIRQD# 1h PIRQB# 5h PIRQF# 2h PIRQC# 6h PIRQG# 3h PIRQD# 7h PIRQH#	3h	RW
11	Reserved	Reserved	0	
10:08	ICR	**Interrupt C Pin Route:** See the IDR description. This field applies to INTC#.	2h	RW
07	Reserved	Reserved	0	
06:04	IBR	**Interrupt B Pin Route:** See the IDR description. This field applies to INTB#.	1h	RW
03	Reserved	Reserved	0	
02:00	IAR	**Interrupt A Pin Route:** See the IDR description. This field applies to INTA#.	0h	RW

14.1.6.8 Offset 3146 - 3147h: D28IR – Device 28 Interrupt Route Register

Table 516. Offset 3146 - 3147h: D28IR – Device 28 Interrupt Route Register

Offset: 3146 - 3147h ***Size:*** *16 bit*

Default Value: 03210h ***Power Well:***

Bits	Name	Description	Reset Value	Access
15	Reserved	Reserved	0	
14:12	IDR	**Interrupt D Pin Route:** Indicates which physical pin on the IICH is connected to the INTD# pin reported for device 28 functions. **Bits Pin Bits Pin** 0h PIRQA# 4h PIRQD# 1h PIRQB# 5h PIRQF# 2h PIRQC# 6h PIRQG# 3h PIRQD# 7h PIRQH#	3h	RW
11	Reserved	Reserved	0	
10:08	ICR	**Interrupt C Pin Route:** See the IDR description. This field applies to INTC#.	2h	RW
07	Reserved	Reserved	0	
06:04	IBR	**Interrupt B Pin Route:** See the IDR description. This field applies to INTB#.	1h	RW
03	Reserved	Reserved	0	
02:00	IAR	**Interrupt A Pin Route:** See the IDR description. This field applies to INTA#.	0h	RW

14.1.6.9 Offset 31FFh: OIC – Other Interrupt Control Register

Table 517. Offset 31FFh: OIC – Other Interrupt Control Register

Offset: 31FFh ***Size:*** *8 bit*

Default Value: *0* ***Power Well:***

Bits	Name	Description	Reset Value	Access
07:02	Reserved	Reserved	0	
01	CEN	**Coprocessor Error Enable:** 0 = FERR# does not generate IRQ13 nor IGNNE#. 1 = If FERR# is low, the IICH generates IRQ13 internally and holds it until an I/O port F0h write. It also drives IGNNE# active.	0	RW
00	AEN	**APIC Enable:** 0 = The internal IOxAPIC is disabled. 1 = Enables the internal IOxAPIC and its address decode.	0	RW

14.1.7 General Configuration Registers

14.1.7.1 Offset 3400 - 3403h: RC – RTC Configuration Register

Table 518. Offset 3400 - 3403h: RC – RTC Configuration Register

Offset: 3400 - 3403h		*Size:* 32 bit		
Default Value: 0		*Power Well:*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:05	Reserved	Reserved	0	
04	UL	**Upper 128 Byte Lock:** 0 = Bytes 38h-3Fh in the upper 128-byte bank of RTC RAM are not locked and can be accessed. Writes are not dropped and reads return any guaranteed data. 1 = Bytes 38h-3Fh in the upper 128-byte bank of RTC RAM are locked and cannot be accessed. Writes are dropped and reads do not return any guaranteed data. Bit reset on system reset.	0	RWLO
03	LL	**Lower 128 Byte Lock:** 0 = Bytes 38h-3Fh in the lower 128-byte bank of RTC RAM are not locked and can be accessed. Writes are not dropped and reads return any guaranteed data. 1 = Bytes 38h-3Fh in the lower 128-byte bank of RTC RAM are locked and cannot be accessed. Writes are dropped and reads do not return any guaranteed data. Bit reset on system reset.	0	RWLO
02	UE	**Upper 128 Byte Enable:** 0 = The upper 128-byte bank of RTC RAM can not be accessed. 1 = The upper 128-byte bank of RTC RAM can be accessed.	0	RW
01:00	Reserved	Reserved	0	

14.1.7.2 Offset 3404 - 3407h: HPTC – High Performance Precision Timer Configuration Register

Table 519. Offset 3404 - 3407h: HPTC – High Performance Precision Timer Configuration Register

Offset: 3404 - 3407h		*Size:* 32 bit		
Default Value: 0		*Power Well:*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:08	Reserved	Reserved	0	
07	AE	**Address Enable:** 0 = The IICH does not decode the High Performance Timer memory address range selected by bits 01:00. 1 = The IICH decodes the High Performance Timer memory address range selected by bits 01:00.	0	RW
06:02	Reserved	Reserved	0	
01:00	AS	**Address Select:** This 2-bit field selects one of four possible memory address ranges for the High Performance Timer functionality. The encodings are: **Bits** **Memory Address Range** 00 FED0_0000h - FED0_03FFh 01 FED0_1000h - FED0_13FFh 10 FED0_2000h - FED0_23FFh 11 FED0_3000h - FED0_33FFh	0	RW

14.1.7.3 Offset 3410 - 3413h: GCS – General Control and Status Register

Table 520. Offset 3410 - 3413h: GCS – General Control and Status Register (Sheet 1 of 2)

Offset: 3410 - 3413h			***Size:*** *32 bit*	
Default Value: *See description below*			***Power Well:***	
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:24	Reserved	Reserved	00h	
23:16	BDS	**BIST_Delay_Sel:** This field determines the amount of time, measured in 125 MHz clocks, waits to deassert the INIT# signal after sending the CPU_RESET_DONE_ACK message. **Notes** 1 This field only has meaning if the BIST_EN bit (Bit 2 in the BUC register) is also set. 2 A value of 00h or 01h in this field is not permitted. 3 A 1 clock variation permitted in the actual-time the INIT# signal goes inactive. 4 This field is in the core well. 5 This field is not reset by a CF9 reset with value 06h. Implementation choice: This register does not need to be reset by any reset.	XX	RW
15:10	Reserved	Reserved	00h	
09	SERM	**Server Error Reporting Mode:** 0 = The IICH is the final target of all errors. The IMCH sends a DO_SERR messages to the IICH for the purpose of generating NMI. 1 = The IMCH is the final target of all errors from PCI Express and NSI. In this mode, if the IICH detects a fatal, non-fatal, or correctable error on NSI or its downstream ports, it sends one of ERR_FATAL, ERR_NONFATAL, or ERR_CORR to IMCH. If the IICH receives an ERR_* message from the downstream port, it sends that message to the IMCH.	0	RW
08:07	Reserved	Reserved	0	
06	FME	**FERR# MUX Enable:** This bit enables FERR# to be a processor break event indication. 0 = IICH does not examine FERR# during a C2, or C4 state as a break event. 1 = IICH examines FERR# during a C2, or C4 state as a break event.	0	RW
05	NR	**No Reboot:** This bit is set when the "No Reboot" strap is sampled high on PWROK. This bit may be set or cleared by software if the strap is sampled low but may not override the strap when it indicates "No Reboot". 0 = The TCO timer does not count down and generate the SMI# on the first timeout, but reboots on the second timeout. 1 = The TCO timer counts down and generates the SMI# on the first timeout, but does not reboot on the second timeout.	Strap	RW
04	AME	**Alternate Access Mode Enable:** 0 = Read-only registers cannot be written, and write-only registers cannot be read. See Section 22.6 for details. 1 = Read-only registers can be written, and write-only registers can be read. See Section 22.6 for details.	0	RW

Table 520. Offset 3410 - 3413h: GCS – General Control and Status Register (Sheet 2 of 2)

Offset: 3410 - 3413h			***Size:*** *32 bit*	
Default Value: *See description below*			***Power Well:***	
Bits	**Name**	**Description**	**Reset Value**	**Access**
03	BBD	**Boot BIOS Destination:** 0 = The top 16 Mbyte of memory below 4 Gbyte (FF00_0000h to FFFF_FFFFh) is accepted by the primary side of the PCI P2P bridge and forwarded to the PCI bus. 1 = This range is not decoded to PCI and the LPC bridge claims these cycles based on the FWH Decode Enable bits. The default value of this bit is determined by a strap (GNT[5]#/ GPIO[17] at rising edge of PWROK, see Section 22.6 for details) allowing systems with corrupted or unprogrammed flash to boot from a PCI device. The value of the strap can be overwritten by software. When this bit is 0, the PCI-to-PCI bridge Memory Space Enable bit does not need to be set (nor any other bits) in order for these cycles to go to PCI. ***Note:*** BIOS enable ranges and the other BIOS protection and update bits associated with the FWH interface have no effect when this bit is 0.	Strap	RW
02	RPR	**Reserved Page Route:** Determines where to send the reserved page registers. These addresses are sent to PCI or LPC for the purpose of generating POST codes. The I/O addresses modified by this field are: 80h, 84h, 85h, 86h, 88h, 8Ch, 8Dh, and 8Eh. 0 = Writes are forwarded to LPC, shadowed within the IICH, and reads are returned from the internal shadow. 1 = Writes are forwarded to PCI, shadowed within the IICH, and reads are returned from the internal shadow. ***Note:*** If some writes are done to LPC/PCI to these I/O ranges, and then this bit is flipped, such that writes now go to the other interface, the reads do not return what was last written. Shadowing is performed on each interface. The aliases for these registers, at 90h, 94h, 95h, 96h, 98h, 9Ch, 9Dh, and 9Eh, are always decoded to LPC.	0	RW
01	Reserved	1 = **Reserved**	0	
00	TSLD	**Top Swap Lock-Down:** 0 = This bit can only be written from 0 to 1 once. BUC.TS can be changed. 1 = Prevents BUC.TS from being changed.	0	RWLO

14.1.7.4 Offset 3414h: BUC – Backed Up Control Register

All bits in this register are in the RTC well and only cleared by RTEST.

Table 521. Offset 3414h: BUC – Backed Up Control Register

Offset: 3414h			***Size:*** *8 bit*	
Default Value: See description below			***Power Well:***	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:03	Reserved	Reserved	0	
02	CBE	**Processor BIST Enable:** 0 = The INIT# signal is not driven active when CPURST# is active. 1 = The INIT# signal is driven active when CPURST# is active. INIT# goes inactive with the same timings as the other CPU Interface signals (hold time after CPURST# inactive). This bit is in the resume well and is reset by RSMRST#, but not PCIRST# nor CF9h writes.	0	RW
01	Reserved	Reserved	0	
00	TS	**Top Swap:** 0 = IICH does not invert A16. 1 = IICH inverts A16 for cycles going to the BIOS space (but not the feature space) in the FWH. If the IICH is strapped for Top-Swap (GNT[6]# is low at rising edge of PWROK), then this bit **cannot** be cleared by software. The strap jumper must be removed and the system rebooted.	Strap	RW

14.1.7.5 Offset 3418 - 341Bh: FD – Function Disable Register

The USB1.1 functions must be disabled from highest function number to lowest. For example, if only two USB1.1 host controllers are wanted, software must disable Function 3. When disabling USB1 host controllers, the USB 2.0 EHCI Structural Parameters Registers must be updated with coherent information in “Number of Companion Controllers” and “N_Ports” fields.

When disabling a function, only the configuration space is disabled. Software must ensure that all functionality within a controller that is not desired (such as memory spaces, I/O spaces, and DMA engines) is disabled prior to disabling the function.

Table 522. Offset 3418 - 341Bh: FD – Function Disable Register (Sheet 1 of 2)

Offset: 3418 - 341Bh			***Size:*** *32 bit*	
Default Value: *00000C12h*			***Power Well:***	
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:20	Reserved	Reserved	0	
19	PE4D	**PCI Express 4 Disable:** 0 = When either the FFPCIE0 or FFPCIE1is cleared, the field is read/write. 1 = PCI Express port B3 is disabled. When either the FFPCIE0 or FFPCIE1 are set, this field becomes a read-only ‘1’. When hidden, the link for this port is put into the “link down” state.	0	RW
18	PE3D	**PCI Express 3 Disable:** 0 = When either the FFPCIE0 or FFPCIE1 is cleared, the field is read/write. 1 = PCI Express port B2 is disabled. When either the FFPCIE0 or FFPCIE1 is set, this field becomes a read-only ‘1’. When hidden, the link for this port is put into the “link down” state.	0	RW

Table 522. Offset 3418 - 341Bh: FD – Function Disable Register (Sheet 2 of 2)

Offset: 3418 - 341Bh **Size:** 32 bit
Default Value: 00000C12h **Power Well:**

Bits	Name	Description	Reset Value	Access
17	PE2D	**PCI Express 2 Disable:** 0 = When the FFPCIE0 is cleared, the field is read/write. 1 = PCI Express port B1 is disabled.	0	RW
16	PE1D	**PCI Express 1 Disable:** 0 = The field is read/write. 1 = PCI Express port B0 is disabled.	0	RW/RO
15	U2D	**USB 2.0 Disable:** 0 = The USB 2.0 host controller is enabled. 1 = The USB 2.0 host controller is disabled.	0	RW
14	LBD	**LPC Bridge Disable:** 0 = The LPC bridge is enabled. 1 = The LPC bridge is disabled. Unlike the other disables in this register, the following additional spaces no longer are decoded by the LPC bridge: • Memory cycles below 16 MBytes (1000000h) • I/O cycles below 64 Kbytes (10000h) • The Internal I/OxAPIC at FEC0_0000 to FECF_FFFF • Memory cycles in the LPC BIOS range below 4 GByte are still decoded when this bit is set, but the aliases at the top of 1 MByte (the E and F segment) are no longer decoded.	0	RW
13:12	Reserved	Reserved	0	
11	U4D	**USB1 #4 Disable:** 0 = When these are cleared, this field becomes read/write defaulting to '0'. 1 = The fourth USB 1.1 controller (ports 6 and 7) is disabled. When either the FFUSB0 or FFUSB1 is set, this field becomes a read-only '1'.	1	RO
10	U3D	**USB1 #3 Disable:** 0 = When this is cleared, this field becomes read/write defaulting to '0'. 1 = The third USB 1.1 controller (ports 4 and 5) is disabled. When the FFUSB0 is set, this field becomes a read-only '1'.	1	RO
09	U2D	**USB1 #2 Disable:** 0 = The second USB 1.1 controller (ports 2 and 3) is enabled. 1 = The second USB 1.1 controller (ports 2 and 3) is disabled.	0	RW
08	U1D	**USB1 #1 Disable:** 0 = When set, the first USB 1.1 controller (ports 0 and 1) is disabled. 1 = When set, the first USB 1.1 controller (ports 0 and 1) is disabled.	0	RW
07	Reserved	Reserved	1	
06	Reserved	Intel® 3100 Chipset BIOS must set this to 1.	1	
05	Reserved	Intel® 3100 Chipset BIOS must set this to 1.	1	
04	Reserved	1 = Reserved	1	
03	SD	**SM Bus Disable:** 0 = The SM Bus controller is enabled. 1 = The SM Bus controller is disabled.	0	RW
02	SAD	**Serial ATA Disable:** 0 = The serial ATA controller is enabled. 1 = The serial ATA controller is disabled.	0	RW
01:00	Reserved	1 = Reserved	1	

14.1.7.6 Offset 341C - 341Fh: PRC – Power Reduction Control Register Clock Gating

Table 523. Offset 341C - 341Fh: PRC – Power Reduction Control Register Clock Gating

Offset: 341C - 341Fh ***Size:*** *32 bit*

Default Value: *0* ***Power Well:***

Bits	Name	Description	Reset Value	Access
31:01	Reserved	Reserved	0	
00	RPSCG	**Root Port Static Clock Gate Enable:** 0 = Static Clock Gating is disabled for the PCI Express Root Port 1 = Static Clock Gating is enabled for the PCI Express Root Port when the corresponding port is disabled in the Function Disable register FD.PE1D, FD.PE2D, FD.PE3D or FD.PE4D. In addition to the PCI Express function disable register, this bit is also qualified by the Root Port Configuration RPC.PC as the physical layer may be required by an enabled port in a x4 configuration.	0	RW

15.0 Device 30, Function 0: PCI to PCI Bridge

15.1 Overview

The PCI Bridge implements buffering and control logic between the PCI and the I/O data bus, and implements the PCI bus arbiter.

15.2 Configuration Registers

15.2.1 Register Summary Table

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values and are read-only. Writes to reserved locations may cause system failure and unpredictable results.

Table 524. PCI to PCI Bridge Register Summary Table (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
PCI Header					
00h	03h	ID	Identifiers Register	244E8086h	RO
04h	05h	CMD	PCI Command Register	0000h	RW, RO
06h	07h	PSTS	Primary Status Register	0010h	RWC, RO
08h	08h	RID	Revision Identification Register	See Description	RO
09h	0Bh	CC	Class Code Register	060401h	RO
0Dh	0Dh	PMLT	Primary Latency Timer Register	00h	RO
0Eh	0Eh	HEADTYP	Header Type Register	*01h*	RO
18h	1Ah	BNUM	Bus Number Register	000000h	RW, RO
1Bh	1Bh	SMLT	Secondary Master Latency Timer Register	00h	RW, RO
1Ch	1Dh	IOBASE_LIMIT	I/O Base and Limit Register	0000h	RW, RO
1Eh	1Fh	SSTS	Secondary Status Register	0280h	RWC, RO
20h	23h	MEMBASE_LIMIT	Memory Base and Limit Register	00000000h	RW, RO
24h	27h	PREF_MEM_ BASE_LIMIT	Prefetchable Memory Base and Limit Register	00010001h	RW, RO
28h	2Bh	PMBU32	Prefetchable Memory Upper 32 Bits Register	00000000h	RW
2Ch	2Fh	PMLU32	Prefetchable Memory Limit Upper 32 Bits Register	00000000h	RW
34h	34h	CAPP	Capability List Pointer Register	50h	RO
3Ch	3Dh	INTR	Interrupt Information Register	0000h	RW, RO
3Eh	3Fh	BCTRL	Bridge Control Register	0000h	RO, RW, RWC

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 524. PCI to PCI Bridge Register Summary Table (Sheet 2 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
Bridge Proprietary Configuration					
40h	41h	SPDH	Secondary PCI Device Hiding Register	00h	RW, RO
44h	47h	DTC	Delayed Transaction Control Register	00000000h	RW, RO
48h	4Bh	BPS	Bridge Proprietary Status Register	00000000h	RO, RWC
4Ch	4Fh	BPC	Bridge Policy Configuration Register	00000000h	RW, RO
PCI Bridge Vendor Capability					
50h	50h	SVCAP	Subsystem Vendor Capability Register	000Dh	RO
54h	54h	SVID	Subsystem Vendor IDs Register	00000000h	RW
Manufacturer's ID					
F8h	F8h	MANID	Manufacturer's ID Register	00010F80h	RO

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

15.2.2 PCI Header

Note: Address locations that are not shown should be treated as Reserved.

15.2.2.1 Offset 00 - 03h: ID – Identifiers Register

Table 525. Offset 00 - 03h: ID – Identifiers Register

Device: 30 ***Function:*** 0
Offset: 00 - 03h ***Size:*** *32 bit*
Default Value: *244E8086h* ***Power Well:***

Bits	Name	Description	Default	Access
31:16	DID	**Device Identification:** This is a 16-bit value assigned to the PCI bridge.	244Eh	RO
15:00	VID	**Vendor Identification:** Indicates Intel.	8086h	RO

15.2.2.2 Offset 04 - 05h: CMD – Command Register

Table 526. Offset 04 - 05h: CMD – Command Register (Sheet 1 of 2)

Device: 30 ***Function:*** 0
Offset: 04 - 05h ***Size:*** *16 bit*
Default Value: *0000h* ***Power Well:***

Bits	Name	Description	Default	Access
15:11	Reserved	Reserved	0	
10	ID	**Interrupt Disable:** The PCI bridge has no interrupts to disable.	0	RO
09	FBE	**Fast Back-to-Back Enable:** Reserved as '0' per the *PCI Express* Base Specification*.	0	RO
08	SEE	**SERR# Enable:** 0 = Disable 1 = Enable the IICH to generate an NMI (or SMI# if NMI routed to SMI#) when the PSTS.SSE bit is set.	0	RW
07	WCC	**Wait Cycle Control:** Reserved as '0' per the *PCI Express Base Specification*.	0	RO

Table 526. Offset 04 - 05h: CMD – Command Register (Sheet 2 of 2)

Device: 30	***Function:*** 0
Offset: 04 - 05h	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:***

Bits	Name	Description	Default	Access
06	PER	**Parity Error Response Enable:** 0 = Disable, ignores parity errors on the PCI bridge. 1 = Enable, sets the SSE bit PSTS.SSE when parity errors are detected on the PCI bridge.	0	RW
05	VPS	**VGA Palette Snoop:** Reserved as '0' per the *PCI Express Base Specification*.	0	RO
04	MWE	**Memory Write and Invalidate Enable:** Reserved as '0' per the *PCI Express Base Specification*.	0	RO
03	SCE	**Special Cycle Enable:** Reserved as '0' per the *PCI Express Base Specification* and *PCI-to-PCI Bridge Specification*.	0	RO
02	BME	**Bus Master Enable:** 0 = Disable, does not allow the bridge to accept cycles from PCI. 1 = Enable, allows the bridge to accept cycles from PCI.	0	RW
01	MSE	**Memory Space Enable:** Controls the response as a target for memory cycles targeting PCI. 0 = Disable 1 = Enable	0	RW
00	IOSE	**I/O Space Enable:** Controls the response as a target for I/O cycles targeting PCI. 0 = Disable 1 = Enable	0	RW

15.2.2.3 Offset 06 - 07h: PSTS – Primary Status Register

Table 527. Offset 06 - 07h: PSTS – Primary Status Register (Sheet 1 of 3)

Device: 30	***Function:*** 0
Offset: 06 - 07h	***Size:*** *16 bit*
Default Value: *0010h*	***Power Well:***

Bits	Name	Description	Default	Access
15	DPE	**Detected Parity Error:** 0 = Parity error not detected. 1 = Indicates that a parity error was detected on the internal I/O data bus. This bit gets set even if the Parity Error Response bit CMD.PER is not set.	0	RWC

Table 527. Offset 06 - 07h: PSTS – Primary Status Register (Sheet 2 of 3)

Device: 30	**Function:** 0
Offset: 06 - 07h	**Size:** 16 bit
Default Value: 0010h	**Power Well:**

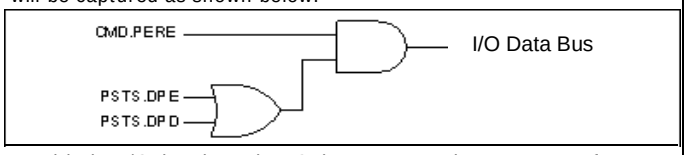
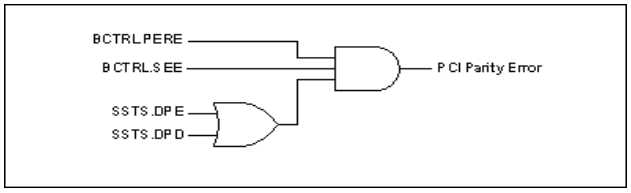
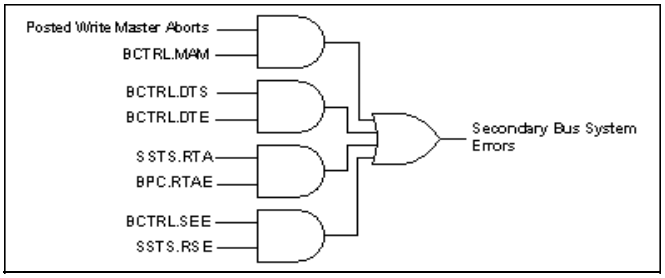
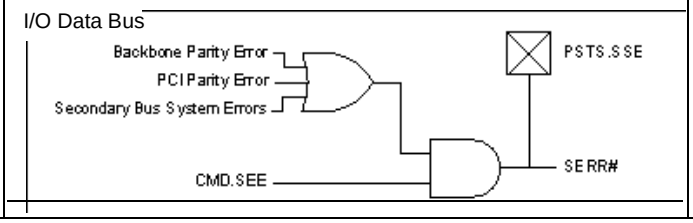
Bits	Name	Description	Default	Access
14	SSE	**Signaled System Error:** See Section 15.2.2.2 to see when this bit is set. Several internal and external sources of the bridge can cause SERR#. The first class of errors is parity errors related to the I/O data bus. The PCI bridge captures generic data parity errors (errors it finds on the I/O data bus) as well as errors returned on I/O data cycles where the bridge was the master. If either of these two conditions is met, and the primary side of the bridge is enabled for parity error response, SERR# will be captured as shown below.  As with the I/O data bus, the PCI bus captures the same sets of errors. The PCI bridge captures generic data parity errors (errors it finds on PCI) as well as errors returned on PCI cycles where the bridge was the master. If either of these two conditions is met, and the secondary side of the bridge is enabled for parity error response, SERR# will be captured as shown below.  The final class of errors is system bus errors. There are three status bits associated with system bus errors, each with a corresponding enable. The diagram capturing system bus error is shown below.  After checking for the three classes of errors, SERR# is generated and PSTS.SSE logs the generation of SERR#, if CMD.SEE (D30:F0:04, bit 8) is set, as shown below.  	0	RWC

Table 527. Offset 06 - 07h: PSTS – Primary Status Register (Sheet 3 of 3)

Device: 30	**Function:** 0
Offset: 06 - 07h	**Size:** 16 bit
Default Value: 0010h	**Power Well:**

Bits	Name	Description	Default	Access
13	RMA	**Received Master Abort:** 0 = No master abort received. 1 = Set when the bridge receives a master abort status from the I/O data bus.	0	RWC
12	RTA	**Received Target Abort:** 0 = No target abort received. 1 = Set when the bridge receives a target abort status from the I/O data bus.	0	RWC
11	STA	**Signaled Target Abort:** 0 = No signaled target abort. 1 = Set when the bridge generates a completion packet with target abort status on the I/O data bus.	0	RWC
10:09	Reserved	Reserved	00	
08	DPD	**Data Parity Error Detected:** 0 = Data parity error not detected. 1 = Set when the bridge receives a completion packet from the I/O data bus from a previous request, and detects a parity error, and CMD.PER is set.	0	RWC
07	Reserved	Reserved	0	
06	Reserved	Reserved	0	
05	Reserved	Reserved	0	
04	CLIST	**Capabilities List:** Capability list exists on the PCI bridge.	1	RO
03	IS	**Interrupt Status:** The PCI bridge does not generate interrupts.	0	RO
02:00	Reserved	Reserved	0	RO

15.2.2.4 Offset 08h: RID – Revision Identification Register

Table 528. Offset 08h: RID – Revision Identification Register

Device: 30	**Function:** 0
Offset: 08h	**Size:** 8 bit
Default Value: 11001XXXb	**Power Well:**

Bits	Name	Description	Default	Access
07:00	RID	**Revision ID:** 8-bit value that indicates the revision number for the PCI bridge. The upper nibble of this register is hardwired to Ch. **Bits Value** 07:04 Ch 03 1b 02:00 Matches RID of LPC bridge in D31, F0. See Section 16.2.1.4 for details.	11001XXXb	RO

15.2.2.5 Offset 09 - 0Bh: CC – Class Code Register

Table 529. Offset 09 - 0Bh: CC – Class Code Register

Device: 30	***Function:*** 0
Offset: 09 - 0Bh	***Size:*** *24 bit*
Default Value: *060401h*	***Power Well:***

Bits	Name	Description	Default	Access
23:16	BCC	**Base Class Code:** Hardwired to 06h. Indicates the device is a bridge device.	06h	RO
15:08	SCC	**Sub-Class Code:** Hardwired to 04h. Indicates the device is a PCI-to-PCI bridge.	04h	RO
07:00	PI	**Programming Interface:** Hardwired to 01h. Indicates the bridge is subtractive decode.	01h	RO

15.2.2.6 Offset 0Dh: PMLT – Primary Latency Timer Register

Table 530. Offset 0Dh: PMLT – Primary Latency Timer Register

Device: 30	***Function:*** 0
Offset: 0Dh	***Size:*** *8 bit*
Default Value: *00h*	***Power Well:***

Bits	Name	Description	Default	Access
07:03	MLC	**Master Latency Count:** Reserved per the *PCI Express Base Specification*, *Rev. 1.0a*.	00h	RO
02:00	Reserved	Reserved	0h	

15.2.2.7 Offset 0Eh: HEADTYP – Header Type Register

Table 531. Offset 0Eh: HEADTYP – Header Type Register

Device: 30	***Function:*** 0
Offset: 0Eh	***Size:*** *8 bit*
Default Value: *01h*	***Power Well:***

Bits	Name	Description	Default	Access
07	Reserved	Reserved.	0b	
06:00	HTYPE	**Header Type:** Identifies the header layout of the configuration space, which is a PCI-to-PCI bridge.	01h	RO

15.2.2.8 Offset 18 - 1Ah: BNUM – Bus Number Register

Table 532. Offset 18 - 1Ah: BNUM – Bus Number Register

Device: 30	***Function:*** 0
Offset: 18 - 1Ah	***Size:*** *24 bit*
Default Value: *000000h*	***Power Well:***

Bits	Name	Description	Default	Access
23:16	SBBN	**Subordinate Bus Number:** Indicates the highest PCI bus number below the bridge. If a Type 1 configuration cycle from the I/O data bus does not fall in the Secondary-to-Subordinate Bus ranges of Device 30, the PCI bridge does not decode the cycle.	00h	RW
15:08	SCBN	**Secondary Bus Number:** Indicates the PCI bus number. Cycles targeting this bus number are converted to type 0 cycles on the PCI bus.	00h	RW
07:00	PBN	**Primary Bus Number:** This is hardwired to 00h for legacy software compatibility.	00h	RO

15.2.2.9 Offset 1Bh: SMLT – Secondary Master Latency Timer Register

This timer controls the amount of time the bridge will burst data on its secondary interface. The bridge will remove GNT# to the master whenever FRAME# is asserted. Also the counter starts counting down from the assertion of FRAME#. Thus the value programmed in the SMLT determines the maximum outbound burst length of the bridge independent of any other outstanding requests.

Table 533. Offset 1Bh: SMLT – Secondary Master Latency Timer Register

Device: 30	***Function:*** 0
Offset: 1Bh	***Size:*** *8 bit*
Default Value: *00h*	***Power Well:***

Bits	Name	Description	Default	Access
07:03	MLTC	**Master Latency Timer Count:** This is a 5-bit value that indicates the number of PCI clocks, in 8-clock increments, that the PCI bridge remains as master of the bus.	00h	RW
02:00	Reserved	Reserved.	00	

15.2.2.10 Offset 1C - 1Dh: IOBASE_LIMIT – I/O Base and Limit Register

Table 534. Offset 1C - 1Dh: IOBASE_LIMIT – I/O Base and Limit Register

Device: 30	***Function:*** 0
Offset: 1C - 1Dh	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:***

Bits	Name	Description	Default	Access
15:12	IOLAL	**I/O Limit Address Limit bits [15:12]:** I/O Base bits corresponding to address lines 15:12 for 4 Kbyte alignment. Bits 11:00 of the 16-bit I/O address are assumed to be padded to FFFh.	0h	RW

Table 534. Offset 1C - 1Dh: IOBASE_LIMIT – I/O Base and Limit Register

Device: 30	***Function:*** 0
Offset: 1C - 1Dh	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:***

Bits	Name	Description	Default	Access
11:08	IOLC	**I/O Limit Address Capability:** Indicates that the bridge does not support 32-bit I/O addressing.	0h	RO
07:04	IOBA	**I/O Base Address:** I/O Base bits corresponding to address lines 15:12 for 4 Kbyte alignment. Bits 11:00 of the 16-bit I/O address are assumed to be padded to 000h.	0h	RW
03:00	IOBC	**I/O Base Address Capability:** Indicates that the bridge does not support 32-bit I/O addressing.	0h	RO

15.2.2.11 Offset 1E - 1Fh: SSTS – Secondary Status Register

Table 535. Offset 1E - 1Fh: SSTS – Secondary Status Register (Sheet 1 of 2)

Device: 30	***Function:*** 0
Offset: 1E - 1Fh	***Size:*** *16 bit*
Default Value: *0280h*	***Power Well:***

Bits	Name	Description	Default	Access
15	DPE	**Detected Parity Error:** 0 = Parity error not detected. 1 = PCI bridge detected an address or data parity error on the PCI bus.	0	RWC
14	RSE	**Received System Error:** 0 = SERR# assertion **not** received. 1 = SERR# assertion is received on the PCI bus.	0	RWC
13	RMA	**Received Master Abort:** 0 = No master abort. 1 = This bit is set whenever the bridge is acting as an initiator on the PCI bus and the cycle is master-aborted. For packets that have completion required, this must also cause a target abort to be returned and sets PSTS.STA. (D30, F0, 06, bit 11)	0	RWC
12	RTA	**Received Target Abort:** 0 = No target abort. 1 = This bit is set whenever the bridge is acting as an initiator on the PCI bus and a cycle is target-aborted on the PCI bus. For packets that have completion required, this event must also cause a target abort to be returned, and sets PSTS.STA.	0	RWC
11	STA	**Signaled Target Abort:** 0 = No target abort. 1 = This bit is set when the bridge is acting as a target on the PCI bus and signals a target abort.	0	RWC
10:09	DEVT	**DEVSEL# Timing:** The bridge is hardwired to medium decode timing.	01	RO
08	DPD	**Data Parity Error Detected:** 0 = The following three conditions below are **not** met. 1 = Set when all of the following three conditions are met: a. The bridge is the initiator on the PCI bus b. PERR# is detected asserted or a parity error is detected internally c. BCTRL.PER (D30, F0,3E, bit 0) is set.	0	RWC
07	FBC	**Fast Back to Back Capable:** Hardwired to '1' indicating that the PCI to PCI target logic is capable of receiving fast back-to-back cycles.	1	RO

Table 535. Offset 1E - 1Fh: SSTS – Secondary Status Register (Sheet 2 of 2)

Device: 30	***Function:*** 0
Offset: 1E - 1Fh	***Size:*** *16 bit*
Default Value: *0280h*	***Power Well:***

Bits	Name	Description	Default	Access
06	Reserved	Reserved	0	
05	66MHZ_CAP	**66 MHz Capable:** The bridge is 33 MHz capable only.	0	RO
04:00	Reserved	Reserved	0	

15.2.2.12 Offset 20 - 23h: MEMBASE_LIMIT – Memory Base and Limit Register

Defines the base and limit, aligned to a 1 Mbyte boundary, of the non-prefetchable memory area of the bridge. Accesses that are within the ranges specified in this register are sent to the PCI bus if CMD.MSE is set. Accesses from PCI that are outside the ranges specified are accepted by the bridge if CMD.BME is set.

Table 536. Offset 20 - 23h: MEMBASE_LIMIT – Memory Base and Limit Register

Device: 30	***Function:*** 0
Offset: 20 - 23h	***Size:*** *32 bit*
Default Value: *00000000h*	***Power Well:***

Bits	Name	Description	Default	Access
31:20	ML	**Memory Limit:** These bits are compared with bits 31:20 of the incoming address to determine the upper 1 Mbyte aligned value (exclusive) of the range. The incoming address must be less than this value.	000h	RW
19:16	Reserved	Reserved	0h	
15:04	MB	**Memory Base:** These bits are compared with bits 31:20 of the incoming address to determine the lower 1 Mbyte aligned value (inclusive) of the range. The incoming address must be greater than or equal to this value.	000h	RW
03:00	Reserved	Reserved	0h	

15.2.2.13 Offset 24 - 27h: PREF_MEM_BASE_LIMIT – Prefetchable Memory Base and Limit Register

Defines the base and limit, aligned to a 1 Mbyte boundary, of the prefetchable memory area of the bridge. Accesses that are within the ranges specified in this register are sent to the PCI bus if CMD.MSE is set. Accesses from the PCI bus that are outside the ranges specified are accepted by the bridge if CMD.BME is set.

Table 537. Offset 24 - 27h: PREF_MEM_BASE_LIMIT – Prefetchable Memory Base and Limit Register

Device: 30	***Function:*** 0
Offset: 24 - 27h	***Size:*** *32 bit*
Default Value: *00010001h*	***Power Well:***

Bits	Name	Description	Default	Access
31:20	PML	**Prefetchable Memory Limit:** These bits are compared with bits 31:20 of the incoming address to determine the upper 1 Mbyte aligned value (exclusive) of the range. The incoming address must be less than this value.	000h	RW
19:16	I64L	**64-bit Indicator:** Indicates support for 64-bit addressing.	1h	RO
15:04	PMB	**Prefetchable Memory Base:** These bits are compared with bits 31:20 of the incoming address to determine the lower 1 Mbyte aligned value (inclusive) of the range. The incoming address must be greater than or equal to this value.	000h	RW
03:00	I64B	**64-bit Indicator:** Indicates support for 64-bit addressing.	1h	RO

15.2.2.14 Offset 28 - 2Bh: PMBU32 – Prefetchable Memory Base Upper 32 Bit Register

Table 538. Offset 28 - 2Bh: PMBU32 – Prefetchable Memory Base Upper 32 Bit Register

Device: 30	***Function:*** 0
Offset: 28 - 2Bh	***Size:*** *32 bit*
Default Value: *00000000h*	***Power Well:***

Bits	Name	Description	Default	Access
31:00	PMBU	**Prefetchable Memory Base Upper Portion:** Upper 32-bits of the prefetchable address base.	00000000h	RW

15.2.2.15 Offset 2C - 2Fh: PMLU32 – Prefetchable Memory Limit Upper 32-Bit Register

Table 539. Offset 2C - 2Fh: PMLU32 – Prefetchable Memory Limit Upper 32-Bit Register

Device: 30	***Function:*** 0
Offset: 2C - 2Fh	***Size:*** *32 bit*
Default Value: *00000000h*	***Power Well:***

Bits	Name	Description	Default	Access
31:00	PMLU	**Prefetchable Memory Limit Upper Portion:** Upper 32-bits of the prefetchable address limit.	00000000h	RW

15.2.2.16 Offset 34h: CAPP – Capabilities List Pointer Register

Table 540. Offset 34h: CAPP – Capabilities List Pointer Register

Device: 30	***Function:*** 0
Offset: 34h	***Size:*** *8 bit*
Default Value: *50h*	***Power Well:***

Bits	Name	Description	Default	Access
07:00	PTR	**Capabilities Pointer:** Indicates that the pointer for the first entry in the capabilities list is at 50h in configuration space.	50h	RO

15.2.2.17 Offset 3C - 3Dh: INTR – Interrupt Information Register

Table 541. Offset 3C - 3Dh: INTR – Interrupt Information Register

Device: 30	***Function:*** 0
Offset: 3C - 3Dh	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:***

Bits	Name	Description	Default	Access
15:08	IPIN	**Interrupt Pin:** The PCI bridge does not assert an interrupt.	00h	RO
07:00	ILINE	**Interrupt Line:** This is a software written value to indicate which interrupt line (vector) the interrupt is connected to. No hardware action is taken on this register. Since the bridge does not generate an interrupt, BIOS should program this value to FFh as per the *PCI Bridge Specification*.	00h	RW

15.2.2.18 Offset 3E - 3Fh: BCTRL – Bridge Control Register

Table 542. Offset 3E - 3Fh: BCTRL – Bridge Control Register (Sheet 1 of 3)

Device: 30	***Function:*** 0
Offset: *3E - 3Fh*	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:***

Bits	Name	Description	Default	Access
15:12	Reserved	Reserved	0000b	
11	DTE	**Discard Timer SERR# Enable:** Controls the generation of SERR# on the primary interface in response to the DTS bit being set. 0 = Do not generate SERR# on a secondary timer discard 1 = Generate SERR# in response to a secondary timer discard	0b	RW
10	DTS	**Discard Timer Status:** 0 = secondary discard timer has not expired. 1 = Set to '1' when the secondary discard timer (see the SDT bit below) expires for a delayed transaction in the hard state.	0b	RWC
09	SDT	**Secondary Discard Timer:** Sets the maximum number of PCI clock cycles that the PCI bridge waits for an initiator on the PCI bus to repeat a delayed transaction request. The counter starts once the delayed transaction data is has been returned by the system and is in a buffer in the PCI bridge. If the master has not repeated the transaction at least once before the counter expires, the PCI bridge discards the transaction from its queue. 0 = The PCI master timeout value is between 2^{15} and 2^{16} PCI clocks 1 = The PCI master timeout value is between 2^{10} and 2^{11} PCI clocks	0b	RW

Table 542. Offset 3E - 3Fh: BCTRL – Bridge Control Register (Sheet 2 of 3)

Device: 30	**Function:** 0
Offset: 3E - 3Fh	**Size:** 16 bit
Default Value: 0000h	**Power Well:**

Bits	Name	Description	Default	Access
08	PDT	**Primary Discard Timer:** PDT is not relevant. This is for Read/Write for software compatibility only.	0b	RW
07	FBE	**Fast Back-to-Back Enable:** Hardwired to 0. The PCI bridge does not generate fast back-to-back cycles on the PCI bus.	0b	RO
06	SBR	**Secondary Bus Reset:** Controls PCIRST# assertion on the PCI bus. 0 = Bridge deasserts PCIRST#. 1 = Bridge asserts PCIRST#. When PCIRST# is asserted, the delayed transaction buffers, posting buffers, and the PCI bus are initialized back to reset conditions. The rest of the Intel® 3100 Chipset and the configuration registers are not affected.	0b	RW
05	MAM	**Master Abort Mode :** Controls the PCI bridge's behavior when a master abort occurs: 0 = Asserts TRDY# on the PCI bus, drives all 1's for reads, and discards data on writes. 1 = Returns a target abort on the PCI bus. Master Abort PCI (non-locked cycles): 0 = Normal completion status is returned on the I/O data bus. 1 = Completer abort completion status is returned on the I/O data bus. All locked reads will return a completer abort completion status on the I/O data bus.	0b	RW
04	V16D	**VGA 16-bit Decode:** 0 = Disables the bridge from decoding 16-bit decoding of VGA I/O address. 1 = Enables the bridge to provide 16-bit decoding of VGA I/O address precluding the decode of VGA alias addresses every 1 Kbyte. This bit requires the VGAE bit in this register be set.	0b	RW
03	VGAE	**VGA Enable:** 0 = Memory and I/O addresses on the secondary interface between the ranges below are claimed. 1 = The bridge forwards the following transactions to the PCI bus regardless of the value of the I/O base and limit registers. The transactions are qualified by CMD.MSE and CMD.IOSE being set. • Memory addresses: 000A0000h-000BFFFFh • I/O addresses: 3B0h-3BBh and 3C0h-3DFh. For the I/O addresses, bits [63:16] of the address must be '0', and bits [15:10] of the address are ignored (i.e., aliased).	0b	RW

Table 542. Offset 3E - 3Fh: BCTRL – Bridge Control Register (Sheet 3 of 3)

Device: 30	**Function:** 0
Offset: 3E - 3Fh	**Size:** 16 bit
Default Value: 0000h	**Power Well:**

Bits	Name	Description	Default	Access
02	IE	**ISA Enable:** This bit only applies to I/O addresses that are enabled by the I/O Base and I/O Limit registers and are in the first 64 Kbyte of PCI I/O space. 0 = Disabled 1 = The bridge blocks any forwarding from primary to secondary I/O transactions addressing the last 768 bytes in each 1 Kbyte block (offsets 100h to 3FFh). In the opposite direction (secondary to primary), I/O transactions are forwarded if they address the last 768 bytes in each 1KByte block	0b	RW
01	SEE	**SERR# Enable:** Controls the forwarding of secondary interface SERR# assertions on the primary interface. 0 = Disabled 1 = The PCI bridge will forward SERR# if the following 3 conditions are met. 1. SERR# is asserted on the secondary interface. 2. This bit is set. 3. CMD.SEE is set.	0b	RW
00	PERE	**Parity Error Response Enable:** 0 = The bridge is disabled for parity error reporting based on parity errors on the PCI bus. 1 = The bridge is enabled for parity error reporting based on parity errors on the PCI bus.	0b	RW

15.2.3 Bridge Proprietary Configuration

Table 543. Bridge Proprietary Configuration Registers Summary

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
40	41h	SPDH	Secondary PCI Device Hide Register	*00h*	RO, RW
44	47h	DTC	Delayed Transaction Control Register	*00000000h*	RO, RW
48	4Bh	BPS	Bridge Proprietary Status Register	*00000000h*	RO, RWC
4C	4Fh	BPC	Bridge Policy Configuration Register	*00000000h*	RO, RW

15.2.3.1 Offset 40 - 41h: SPDH – Secondary PCI Device Hiding Register

This register allows software to hide the PCI devices, either plugged into slots or on the motherboard.

Table 544. Offset 40 - 41h: SPDH – Secondary PCI Device Hiding Register

Device: 30	***Function:*** 0
Offset: 40 - 41h	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:***

Bits	Name	Description	Default	Access
15:08	Reserved	Reserved	0	
07	HD7	**Hide Device 7:** Same as bit 0 of this register, except for device 7 (AD[23]).	0	RW
06	HD6	**Hide Device 6:** Same as bit 0 of this register, except for device 6 (AD[22]).	0	RW
05	HD5	**Hide Device 5:** Same as bit 0 of this register, except for device 5 (AD[21]).	0	RW
04	HD4	**Hide Device 4:** Same as bit 0 of this register, except for device 4 (AD[20]).	0	RW
03	HD3	**Hide Device 3:** Same as bit 0 of this register, except for device 3 (AD[19]).	0	RW
02	HD2	**Hide Device 2:** Same as bit 0 of this register, except for device 2 (AD[18]).	0	RW
01	HD1	**Hide Device 1:** Same as bit 0 of this register, except for device 1 (AD[17]).	0	RW
00	HD0	**Hide Device 0:** 0 = When cleared, PCI configuration cycles for this device are not affected. 1 = This bit hides device 0 on the PCI bus by driving IDSEL low for PCI configuration cycles to that device. AD[16] is used as IDSEL for device 0.	0	RW

15.2.3.2 Offset 44 - 47h: DTC – Delayed Transaction Control Register

Table 545. Offset 44 - 47h: DTC – Delayed Transaction Control Register (Sheet 1 of 2)

Device: 30	***Function:*** 0
Offset: 44 - 47h	***Size:*** *32 bit*
Default Value: *00000000h*	***Power Well:***

Bits	Name	Description	Default	Access
31	DDT	**Discard Delayed Transactions:** 0 = This bit is cleared by the PCI bridge when the delayed transaction queues are empty and have returned to an idle state. Software sets this bit and polls for its completion. 1 = When set, the PCI bridge discards any delayed transactions it has logged. This includes transactions in the pending queue, and any transactions in the active queue, whether in the hard or soft DT state. The prefetchers are disabled and return to an idle state. ***Note:*** If a transaction is running on the PCI bus at the time this bit is set, that transaction continues until either the PCI master disconnects (by deasserting FRAME#) or the PCI bridge disconnects (by asserting STOP#). This bit is cleared by the PCI bridge when the delayed transaction queues are empty and have returned to an idle state. Software sets this bit and polls for its completion.	0	RW
30	BDT	**Block Delayed Transactions:** 0 = The PCI bridge accepts incoming transactions. 1 = The PCI bridge does not accept incoming transactions which results in delayed transactions. It blindly retries these cycles by asserting STOP#. All postable cycles (memory writes) are still accepted.	0	RW
29:08	Reserved	Reserved	0	
07:06	MDT	**Maximum Delayed Transactions:** Controls the maximum number of delayed transactions that the PCI bridge runs. Encodings are: **Bits** **Value** 00 2 Active, 5 pending 01 2 active, no pending 10 1 active, no pending 11 Reserved	00	RW
5	Reserved	Reserved	0	
04	AFADE	**Auto Flush After Disconnect Enable:** 0 = The PCI bridge retains any fetched data until required to discard by producer/consumer rules. 1 = The PCI bridge flushes any prefetched data after either the PCI master (by deasserting FRAME#) or the PCI bridge (by asserting STOP#) disconnects the PCI transfer.	0	RW
03	NP	**Never Prefetch:** 0 = Prefetch is enabled. 1 = Only fetches a single Dword and does not enable prefetching, regardless if the command being a Memory read (MR), Memory read line (MRL), or Memory read multiple (MRM).	0	RW

Table 545. Offset 44 - 47h: DTC – Delayed Transaction Control Register (Sheet 2 of 2)

Device: 30	**Function:** 0
Offset: 44 - 47h	**Size:** 32 bit
Default Value: 00000000h	**Power Well:**

Bits	Name	Description	Default	Access
02	MRMPD	**Memory Read Multiple Prefetch Disable:** 0 = MRM commands fetch multiple cache lines as defined by the prefetch algorithm. 1 = Memory read multiple (MRM) commands fetch only up to a single 64 byte aligned cache line.	0	RW
01	MRLPD	**Memory Read Line Prefetch Disable:** 0 = MRL commands fetch multiple cache lines as defined by the prefetch algorithm. 1 = Memory read line (MRL) commands fetch only up to a single 64-byte aligned cache line.	0	RW
00	MRPD	**Memory Read Prefetch Disable:** 0 = MR commands fetch up to a 64-byte aligned cache line. 1 = Memory read (MR) commands fetch only a single Dword.	0	RW

15.2.3.3 Offset 48 - 4Bh: BPS – Bridge Proprietary Status Register

Table 546. Offset 48 - 4Bh: BPS – Bridge Proprietary Status Register (Sheet 1 of 2)

Device: 30	**Function:** 0
Offset: 48 - 4Bh	**Size:** 32 bit
Default Value: 00000000h	**Power Well:**

Bits	Name	Description	Default	Access
31:17	Reserved	Reserved	0	
16	PAD	**PERR# Assertion Detected:** This bit is set by hardware whenever the PERR# pin is asserted on the rising edge of PCI clock. This includes cases in which the Intel® 3100 Chipset is the agent driving PERR#. It remains asserted until cleared by software writing a '1' to this location. When enabled by the PERR#-to-SERR# Enable bit (BPC.PSE register Table 547 on page 571), a '1' in this bit generates an internal SERR# and be a source for the NMI logic. This bit can be used by software to determine the source of a system problem.	0	RWC
15:07	Reserved	Reserved	0	

Table 546. Offset 48 - 4Bh: BPS – Bridge Proprietary Status Register (Sheet 2 of 2)

Device: 30	**Function:** 0
Offset: 48 - 4Bh	**Size:** 32 bit
Default Value: 00000000h	**Power Well:**

Bits	Name	Description	Default	Access
06:04	NPT	**Number of Pending Transactions:** This read-only indicator tells debug software how many transactions are in the pending queue. Possible values are: **Bits** **Value** 000 No pending transactions 001 1 pending transaction 010 2 pending transactions 011 3 pending transactions 100 4 pending transactions 101 5 pending transactions 110 - 111Reserved This field is only valid if DTC.MDT is '00'.	00	RO
03:02	Reserved	Reserved	0	
01:00	NAT	**Number of Active Transactions:** This read-only indicator tells debug software how many transactions are in the active queue. Possible values are: **Bits** **Value** 00 No active transactions 01 1 active transaction 10 2 active transactions 11 Reserved	00	RO

15.2.3.4 Offset 4C - 4Fh: BPC – Bridge Policy Configuration Register

Table 547. Offset 4C - 4Fh: BPC – Bridge Policy Configuration Register (Sheet 1 of 2)

Device: 30	**Function:** 0
Offset: 4C - 4Fh	**Size:** 32 bit
Default Value: 00000000h	**Power Well:**

Bits	Name	Description	Default	Access
31:07	Reserved	Reserved	0	
06	PSE	**PERR#-to-SERR# Enable:** 0 = A '1' in the PERR# Assertion status bit (in the Bridge Proprietary Status register) does **not** result in an internal SERR# assertion on the primary side of the bridge. It also enabled by the SERR# Enable bit in the primary Command register. SERR# is a source of NMI. 1 = A '1' in the PERR# Assertion status bit (in the Bridge Proprietary Status register) results in an internal SERR# assertion on the primary side of the bridge. It also enabled by the SERR# Enable bit in the primary Command register. SERR# is a source of NMI.	0	RW
05	SDTT	**Secondary Discard Timer Testmode:** 0 = The secondary discard timer expiration is defined in BCTRL.SDT. 1 = The secondary discard timer expires after 128 PCI clocks.	0	RW
04:03	Reserved	Reserved	0	

Table 547. Offset 4C - 4Fh: BPC – Bridge Policy Configuration Register (Sheet 2 of 2)

Device: 30	**Function:** 0
Offset: 4C - 4Fh	**Size:** 32 bit
Default Value: 00000000h	**Power Well:**

Bits	Name	Description	Default	Access
02	PDE	**Peer Decode Enable:** 0 = The PCI bridge assumes that all memory cycles target main memory, and all I/O cycles are not claimed. 1 = The PCI bridge performs peer decode on any memory or I/O cycle from PCI that falls outside of the memory and I/O window registers.	0	RW
01	Reserved	Reserved	0	
00	RTAE	**Received Target Abort SERR# Enable:** 0 = The PCI bridge does **not** report SERR# when SSTS.RTA is set, and CMD.SEE is set. 1 = The PCI bridge reports SERR# when SSTS.RTA is set, and CMD.SEE is set.	0	RW

15.2.4 PCI Bridge Vendor Capability

15.2.4.1 Offset 50h: SVCAP – Subsystem Vendor Capability Register

Table 548. Offset 50h: SVCAP – Subsystem Vendor Capability Register

Device: 30	**Function:** 0
Offset: 50h	**Size:** 16 bit
Default Value: 000Dh	**Power Well:**

Bits	Name	Description	Default	Access
15:08	NEXT	**Next Capability:** Value of 00h indicates this is the last item in the list.	00h	RO
07:00	CID	**Capability Identifier:** Value of 0Dh indicates this is a PCI bridge subsystem vendor capability.	0Dh	RO

15.2.4.2 Offset 54h: SVID – Subsystem Vendor IDs Register

Table 549. Offset 54h: SVID – Subsystem Vendor IDs Register

Device: 30	***Function:*** 0
Offset: 54h	***Size:*** *32 bit*
Default Value: *00000000h*	***Power Well:***

Bits	Name	Description	Default	Access
31:16	SID	**Subsystem Identifier:** Indicates the subsystem as identified by the vendor. This field is written once and is locked until a bridge reset occurs (not the PCI bus reset).	0000h	RW
15:00	SVID	**Subsystem Vendor Identifier:** Indicates the manufacturer of the subsystem. This field is written once and is locked until a bridge reset occurs (not the PCI bus reset).	0000h	RW

15.2.5 Manufacturing Information

15.2.5.1 Offset F8h: MANID – Manufacturer's ID Register

Table 550. Offset F8h: MANID – Manufacturer's ID Register

Device: 30	***Function:*** 0
Offset: F8h	***Size:*** *32 bit*
Default Value: *00010F80h*	***Power Well:***

Bits	Name	Description	Default	Access
31:24	Reserved	Reserved	00h	RO
23:16	SID	**Stepping Identifier:** This field increments for each stepping of the Intel® 3100 Chipset. A single Stepping ID can be implemented that is readable from all functions in the chip. ***Note:*** This field can be used by software to differentiate steppings when the Revision ID may not change.	01h = A1	RO
15:08	MID	**Manufacturing Identifier:** 0Fh = Intel	0Fh	RO
07:00	Reserved	Reserved	80h	

15.3 PCI 32/33 Bus Interface

The PCI interface provides a 32-bit 33 MHz, Intel® 3100 Chipset is based on the *PCI Local Bus Specification, Revision 2.3*. All PCI signals, except PME#, are 5 V tolerant. The PCI arbiter supports up to two external PCI bus masters. The

15.4 PCI Bridge as an Initiator

The bridge initiates cycles on the PCI bus on behalf of the I/O data bus when granted access by the PCI arbiter. The bridge generates the cycle types shown in Table 551.

Table 551. PCI Bridge Initiator Cycle Types (Sheet 1 of 2)

Command	C/BE#	Notes
I/O Read	2h	Non-posted
I/O Write	3h	Non-posted
Memory Read	6h	Non-posted

Table 551. PCI Bridge Initiator Cycle Types (Sheet 2 of 2)

Command	C/BE#	Notes
Memory Write	7h	Posted
Configuration Read	Ah	Non-posted
Configuration Write	Bh	Non-posted
Special Cycles	1h	Posted

15.4.1 Memory Reads and Writes

The bridge bursts memory writes on the PCI bus that are received as a single packet from the I/O data bus.

15.4.2 I/O Reads and Writes

The bridge generates single Dword I/O read and write cycles. When the cycle completes on the PCI bus, the bridge generates a corresponding completion on the I/O data bus. If the cycle is retried, the cycle is kept in the downbound queue and may be passed by a postable cycle.

15.4.3 Configuration Reads and Writes

The bridge generates single Dword configuration read and write cycles. When the cycle completes on the PCI bus, the bridge generates a corresponding completion. If the cycle is retried, the cycle is kept in the downbound queue and may be passed by a postable cycle.

15.4.4 Locked Cycles

The bridge propagates locks from the I/O data bus as defined in the *PCI Specification*. The PCI Bridge implements bus lock, which means the arbiter will not grant the bus to any agent except the I/O data bus while it is locked.

If a locked read results in a target or master abort, the lock is not established, as defined in the *PCI Specification*.

15.4.5 Target/Master Aborts

When a cycle initiated by the bridge is master/target aborted, the bridge does not reattempt the same cycle. For multiple Dword cycles, the bridge increments the address and attempts the next Dword of the transaction. For all non-postable cycles, a target abort response packet is returned for each Dword that was master or target aborted on the PCI bus. The bridge drops posted writes that abort.

15.4.6 Secondary Master Latency Timer

Master Latency Timer (MLT) is supported and details can be referred from the PCI Specification and the SLT register description found under Chapter 15.0, “Offset 1Bh: SMLT – Secondary Master Latency Timer Register”.

15.4.7 Dual Address Cycle (DAC)

The bridge issues full 64-bit dual address cycles for device memory-mapped registers above 4 Gbyte.

15.4.8 Memory and I/O Decode to PCI

The PCI Bridge in the Intel® 3100 Chipsetis a subtractive decode agent, which follows the rules below when forwarding a cycle from the I/O data bus to the PCI interface:

- The PCI Bridge will positively decode any memory/I/O address within its window registers, assuming CMD.MSE is set for memory windows and CMD.IOSE is set for I/O windows.
- The PCI Bridge will subtractively decode any 64-bit memory address not claimed by another agent, assuming CMD.MSE is set or Subtractive Decode Policy bit is cleared.
- The PCI bridge will subtractively decode any 16-bit I/O address not claimed by another agent assuming CMD.IOSE is set or Subtractive Decode Policy bit is cleared.
- If BCTRL.IE is set, the PCI Bridge will not positively forward from primary to secondary ranges in the I/O window, as defined in the *PCI Specification*. The PCI Bridge will still take them subtractively assuming the above rules.
- If BCTRL.VAE is set, the PCI Bridge will positively forward from primary to secondary I/O and memory ranges as called out in the *PCI Bridge Specification*, assuming the above rules are met.

15.5 PCI Bridge as a Target

The PCI bridge accepts all cycles that are outside the window ranges defined by the bridge, if CMD.BME is set. The bridge responds.

All cycles are decoded by the bridge in medium time.

15.5.1 Memory Writes

- The bridge does not support non-linear burst orders, and asserts STOP# with the first data transfer.
- Bursts are disconnected with data when either command or data space is full. The bus is not held in wait states.
- Bursts are always terminated on 4 KByte boundaries.
- The bridge breaks large writes from PCI into 64-byte aligned packets for transferring on the I/O data bus
- When the inbound data or command queues are full at the start of the write cycle, the bridge immediately retries the cycle.
- The bridge performs no upstream write combining.

15.5.2 Terminology

- **Active DT** – a delayed transaction has been launched to memory. Active DTs can be in one of two states. “Hard”, which means no TRDY# has been returned, or “Soft”, which means one TRDY# has been returned.
- **Pending DT –** a delayed transaction has been logged internally, but has no data buffer allocated to it.
- **Hard DT –** data in an active DT has not yet been returned to a PCI master (no TRDY# assertion). DTs in this state only discard data when a delayed transaction timeout occurs (via BCTRL.SDT), or the bridge is told to flush transactions (via BPC.DDT).
- **Soft DT –** data in an active DT has returned one TRDY#. DTs in this state discard data based upon the data retention policy (see Section 15.5.2.2), or the bridge is told to flush transactions (via BPC.DDT).

15.5.2.1 PCI Bus Usage Policy

The PCI bridge never holds agents in wait states awaiting data. When a PCI master connects, if at least one Dword of data is available, TRDY# is asserted. If no data is available, the transaction is immediately retried. If data that was in the DT buffer drains, the bridge immediately disconnects with the last data.

15.5.2.2 Data Retention Policy

Data is retained for up to the value specified in BCTRL.SDT in the hard DT state. When data is returned, a counter starts. If the originating PCI agent has not come back for that data within this time, it is discarded.

When some data has been returned to a PCI agent, as defined by TRDY# being driven by the bridge, data is now in the soft DT state and may be discarded at any time. The PCI bridge attempts to maintain that data forever, but discards on the following conditions:

- Peer-to-Peer Write on the PCI bus

This rule ensures that two masters who are sharing a data resource keep data coherent. If agent A wrote to a memory location, and B was reading from that same location, the bridge may have fetched some stale data (the fetch-ahead was performed while the write was in flight). If agent A writes to agent B to say that new data is available, the bridge needs to know this to discard any potentially stale data.

- Outbound Cycle (read, write, I/O, memory, or configuration)

  The processor may have updated a region of memory after the bridge fetched it. It may then indicate to the device on PCI that new data is available. While this is typically a write to the device, there is no rule that says it could not have been a read, so the bridge is discarded.

- Read by another master to the same 4 K page as already enabled for fetching

  This is to ensure that two agents reading from the same 4 K page do not get different data. If the bridge had prefetched data for one agent, the data could have been changed by the processor, and the data does not look the same to another agent.

- Writes by any master to the same 4 K page as already enabled for fetching

  This is to ensure that writes by an agent are reflected in the following read. The bridge does not write merge prefetched data, so the data in the buffer could get stale.

- Read by the same master to an address not in the same 4 K page

  This rule is a trigger that the agent on PCI has finished with the current request, and is moving on to another request. Thus, the value in maintaining this stream is limited, so the bridge discards.

- DTs in the pending queue

  If any transactions are in the pending queue, the bridge moves to those transactions. This is to ensure that all agents make forward progress, as opposed to only a select few that happened to get in “first”.

  When DTs are in the pending queue, and there are two active transactions in the soft DT state, the least recently accessed transaction must be discarded first.

Special considerations must be used when flagging a transaction for discard. A transaction must be flagged for discard in the soft DT state when any of the above events occur, even if the first TRDY# had not yet been delivered. In other words, if any of the above discard rules happens, as soon as a transaction moves from the hard DT state to the soft DT state, it must be discarded.

The reasoning for this is as follows: Assume an agent performed a MRM, which the bridge retries to establish a delayed transaction. As part of this, the bridge fetches 128 B. Before a TRDY# is returned, but after the data has been put in the PCI bridge buffer, the processor updates the upper 64 B and writes a flag to the device. In this case, the upper 64 B is stale. By PCI rule, the PCI agent must come back for a retried transaction, and does so, fetching the first 64 B.

The agent now disconnects, and reconnects to get the other 64 B. If the bridge does not discard the upper 64 B and refetch it, stale data will have been delivered. Thus, the bridge must log the fact that the downstream processor access occurred to know to discard the transaction as soon as there is a PCI disconnect.

15.6 Parity Error Detection and Generation

PCI parity errors can be detected and reported. The following behavioral rules apply:

- When a parity error is detected on PCI, the bridge sets the SSTS.DPE bit.

- If the bridge is a master and the BCTRL.PERE bit and one of the parity errors defined below is detected on PCI, then the bridge sets SSTS.DPD and also generates an internal SERR#.
  - — During a write cycle, the PERR# signal is active; or
  - — A data parity error is detected while performing a read cycle.
- If an address or command parity error is detected on PCI and CMD.SEE, BCTRL.PERE, and BCTRL.SEE are all set, the bridge sets the PSTS.SSE and generates an internal SERR#.
- If the PSTS.SSE is set because of an address parity error and the CMD.SEE is set, the bridge generates an internal SERR#.
- When bad parity is detected from the I/O data bus, bad parity is driven on all data the bridge.
- When an address parity error is detected on PCI, the PCI Bridge never claims the cycle. This is a slight deviation from the *PCI Bridge Specification*, which says that a cycle should be claimed if BCTRL.PERE is not set. However, the internal I/O data bus does not have a concept of address parity error, so claiming the cycle could result in the rest of the system seeing a bad transaction as a good transaction.

15.7 PCI Reset

The PCIRST# pin is generated under two conditions:

- PLTRST# active
- BCTRL.SBR set to ‘1’

The PCIRST# pin is in the resume well. It must only be tied to PCI bus agents and not to other agents in the system.

16.0 Device 31, Function 0: LPC Interface

16.1 Overview

The LPC bridge function resides in PCI Device 31, Function 0. This function contains many other functional units, such as DMA and Interrupt controllers, Timers, Power Management, System Management, GPIO, RTC, and LPC Configuration Registers.

16.2 Configuration Registers (LPC I/F – D31, F0)

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 552. LPC I/F – D31, F0 Configuration Registers Summary Table (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
PCI Configuration Registers					
00h	03h	ID	Vendor Identification Register	2670h 8086h	RO
04h	05h	CMD	Device Command Register	0007h	RO, RW
06h	07h	STS	Status Register	0200h	RWC, RO
08h	08h	RID	Revision ID Register	01h	RW Once
09h	0Bh	CC	Class Code Register	060100h	RO
0Dh	0Dh	MLT	Master Latency Timer Register	00h	RO
0Eh	0Eh	HTYPE	Header Type Register	80h	RO
2Ch	2Fh	SID	Subsystem Identifiers Register	0000_0000	RW Only
ACPI/GPIO Configuration Registers					
40h	40h	ABASE	ACPI Base Address Register	00000001h	RO, RW
44h	44h	ACTL	ACPI Control Register	00h	RO, RW
48h	48h	GBA	GPIO Base Address Register	00000001h	RO, RW
4Ch	4Ch	GC	GPIO Control Register	00h	RO, RW
Interrupt Configuration Registers					
60h	60h	PARC	PIRQA Routing Control Register	80h	RW
61h	61h	PBRC	PIRQB Routing Control Register	80h	RW
62h	62h	PCRC	PIRQC Routing Control Register	80h	RW
63h	63h	PDRC	PIRQD Routing Control Register	80h	RW
64h	64h	SCNT	Serial IRQ Control Register	10h	RO, RW
68h	68h	PERC	PIRQE Routing Control Register	80h	RW
69h	69h	PFRC	PIRQF Routing Control Register	80h	RW
6Ah	6Ah	PGRC	PIRQG Routing Control Register	80h	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 552. LPC I/F – D31, F0 Configuration Registers Summary Table (Sheet 2 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
6Bh	6Bh	PHRC	PIRQH Routing Control Register	80h	RW
LPC I/O Configuration Registers					
80h	81h	IOD	I/O Decode Ranges Register	0000h	RO, RW
82h	83h	IOE	I/O Enables Register	0000h	RO, RW
84h	85h	LG1	LPC Generic Decode Range 1 Register	0000h	RO, RW
88h	88h	LG2	LPC Generic Decode Range 2 Register	0000h	RO, RW
Power Management Configuration Registers: Refer to Chapter 22.0, "Power Management"					
FWH Configuration Registers					
D0h	D3h	FS1	FWH ID Select 1 Register	00112233h	RO, RW
D4h	D5h	FS2	FWH ID Select 2 Register	4567h	RO, RW
D8h	DBh	FDE	FWH Decode Enable Register	FFCFh	RO, RW
DCh	DCh	BC	BIOS Control Register	00h	RO, RW Only
Root Complex Register Block Configuration Register					
F0h	F0h	RCBA	Root Complex Base Address Register	00000000h	RO, RW
Manufacturing Information Register					
F8h	F8h	MANID	Manufacturer's ID Register	00010F80h	RO

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

16.2.1 PCI Configuration Registers

16.2.1.1 Offset 00 - 03h: ID – Vendor Identification Register

Table 553. Offset 00 - 03h: ID – Vendor Identification Register

Device: 31 ***Function:*** 0

Offset: 00 - 03h ***Size:*** *32 bit*

Default Value: *26708086h* ***Power Well:*** Core

Bits	Name	Description	Default	Access
31:16	DID	**Device Identification:** This is hardcoded to 2670.	2670h	RO
15:00	VID	**Vendor Identification:** This 16-bit value is assigned to Intel. Intel VID = 8086h	8086h	RO

16.2.1.2 Offset 04 - 05h: CMD – Device Command Register

Table 554. Offset 04 - 05h: CMD – Device Command Register

Device: 31	***Function:*** 0
Offset: 04 - 05h	***Size:*** *16 bit*
Default Value: *0007h*	***Power Well:*** Core

Bits	Name	Description	Default	Access
15:10	Reserved	Reserved	00h	
09	FBE	**Fast Back to Back Enable:** Hardwired to '0' as per *PCI Express* Base Specification*.	0	RO
08	SEE	**SERR# Enable:** 0 = LPC bridge does not generates SERR# 1 = LPC bridge generates SERR#	0	RW
07	WCC	**Wait Cycle Control:** Hardwired to '0' as per *PCI Express Base Specification*.	0	RO
06	PERE	**Parity Error Response Enable:** 0 = No action is taken when detecting a parity error. 1 = Enables the LPC bridge to respond to parity errors detected on IICH interface.	0	RW
05	VGA_PSE	**VGA Palette Snoop:** Hardwired to '0' as per *PCI Express Base Specification*.	0	RO
04	MWIE	**Memory Write and Invalidate Enable:** Hardwired to '0'as per *PCI Express Base Specification*.	0	RO
03	SCE	**Special Cycle Enable:** Hardwired to '0' as per *PCI Express Base Specification*.	0	RO
02	BME	**Bus Master Enable:** Bus Masters cannot be disabled.	1	RO
01	MSE	**Memory Space Enable:** Memory space cannot be disabled on LPC.	1	RO
00	IOSE	**I/O Space Enable:** I/O space cannot be disabled on LPC.	1	RO

16.2.1.3 Offset 06 - 07h: STS – Status Register

Table 555. Offset 06 - 07h: STS – Status Register (Sheet 1 of 2)

Device: 31	***Function:*** 0
Offset: 06 -07h	***Size:*** *16 bit*
Default Value: *0200h*	***Power Well:*** Core

Bits	Name	Description	Default	Access
15	DPE	**Detected Parity Error:** Set when the LPC bridge detects an internal parity error. This bit gets set even if CMD.PERE is not set. 0 = Parity Error Not detected 1 = Parity Error detected	0	RWC
14	SSE	**Signaled System Error:** Set when the LPC bridge signals a system error to the internal SERR# logic.	0	RWC
13	RMA	**Received Master Abort:** 0 = Unsupported request status not received 1 = The bridge received a completion with unsupported request status from the IICH	0	RWC
12	RTA	**Received Target Abort:** 0 = Completion abort not received 1 = Completion with completion abort received from the IICH	0	RWC

Table 555. Offset 06 - 07h: STS – Status Register (Sheet 2 of 2)

Device: 31	**Function:** 0
Offset: 06 -07h	**Size:** *16 bit*
Default Value: *0200h*	**Power Well:** Core

Bits	Name	Description	Default	Access
11	STA	**Signaled Target Abort:** 0 = Target abort not generated on the IICH 1 = LPC bridge generated a completion packet with target abort status on the IICH	0	RWC
10:09	DTS	**DEVSEL# Timing Status:** 01 = Medium Timing	01	RO
08	DPD	**Data Parity Error Detected:** 0 = All conditions listed below NOT met 1 = Set when all three of the following conditions are met: • LPC bridge receives a completion packet from the IICH from a previous request • Parity error has been detected (D31, F0, 06, bit 15) • PCICMD.PERE bit (D31, F0, 04, bit 6) is set	0	RWC
07:00	Reserved	Reserved	0	

16.2.1.4 Offset 08h: RID – Revision ID Register

Writing to this register controls what is reported in all of the RID registers in the component. The value written does not get directly loaded in this register. However, the value is checked to determine which value to report.

Note: Once written, additional writes to this register must not have any affect until a core-well reset occurs. BIOS must always write to this register in order to guarantee that the functionality is locked.

Table 556. Offset 08h: RID – Revision ID Register

Device: 31	**Function:** 0
Offset: 08h	**Size:** *8 bit*
Default Value: *01h*	**Power Well:** Core

Bits	Name	Description	Default	Access
07:00	RID	**Revision ID:** Indicates the part revision	01h	RWO

16.2.1.5 Offset 09 - 0Bh: CC – Class Code Register

Table 557. Offset 09 - 0Bh: CC – Class Code Register

Device: 31	***Function:*** 0
Offset: 09 - 0Bh	***Size:*** *24 bit*
Default Value: *060100h*	***Power Well:*** Core

Bits	Name	Description	Default	Access
23:16	BCC	**Base Class Code:** indicates the type of device for the LPC bridge. 06h = Bridge device.	06h	RO
15:08	SCC	**Sub-Class Code:** Indicates the category of bridge for the LPC bridge. 01h = PCI-to-ISA bridge.	01h	RO
07:00	PI	**Programming Interface:** The LPC bridge has no programming interface.	00h	RO

16.2.1.6 Offset 0Dh: MLT – Master Latency Timer Register

Table 558. Offset 0Dh: MLT – Master Latency Timer Register

Device: 31	***Function:*** 0
Offset: 0Dh	***Size:*** *8 bit*
Default Value: *00h*	***Power Well:*** Core

Bits	Name	Description	Default	Access
07:03	MLC	**Master Latency Count:** Reserved per *PCI Express Base Specification*.	0h	
02:00	Reserved	Reserved	0	

16.2.1.7 Offset 0Eh: HTYPE – Header Type Register

Table 559. Offset 0Eh: HTYPE – Header Type Register

Device: 31	***Function:*** 0
Offset: 0Eh	***Size:*** *8 bit*
Default Value: *80h*	***Power Well:*** Core

Bits	Name	Description	Default	Access
07	MFD	**Multi-function Device:** This bit is hardwired to '1' to indicate a multi-function device.	1	RO
06:00	HTYPE	**Header Type:** Identifies the header layout of the configuration space, which is a generic device.	00h	RO

16.2.1.8 Offset 2C - 2Fh: SID – Subsystem Identifiers Register

This register is initialized to logic 0 by the assertion of PLTRST#. This register can be written only once after PLTRST# deassertion.

Table 560. Offset 2C - 2Fh: SID – Subsystem Identifiers Register

Device: 31	***Function:*** 0
Offset: 2Ch	***Size:*** *32 bit*
Default Value: *0000_0000*	***Power Well:*** Core

Bits	Name	Description	Default	Access
31:16	SSID	**Subsystem ID:** This is written by BIOS. No hardware action taken on this value.	0000h	RWO
15:00	SSVID	**Subsystem Vendor ID:** This is written by BIOS. No hardware action taken on this value.	0000h	RWO

16.2.2 ACPI/GPIO Configuration Registers

16.2.2.1 Offset 40 - 43h: ABASE – ACPI Base Address Register

Sets base address for ACPI I/O registers and TCO I/O registers. Can be mapped anywhere in the 64 K I/O space on 128-byte boundaries.

Table 561. Offset 40 - 43h: ABASE – ACPI Base Address Register

Device: 31	***Function:*** 0
Offset: 40h	***Size:*** *32 bit*
Default Value: *00000001h*	***Power Well:*** Core

Bits	Name	Description	Default	Access
31:16	Reserved	Reserved.	0000h	
15:07	BSA	**Base Address**: This field provides the 128 bytes of I/O space for ACPI and TCO logic. This is placed on a 128 byte boundary.	00h	RW
06:01	Reserved	Reserved.	00h	
00	RTE	**Resource Type Indicator**: Hardwired 1 to indicate I/O space.	1	RO

16.2.2.2 Offset 44 - 47h: ACTL – ACPI Control Register

Table 562. Offset 44 - 47h: ACTL – ACPI Control Register

Device: 31	**Function:** 0
Offset: 44h	**Size:** *8 bit*
Default Value: *00h*	**Power Well:** Core

Bits	Name	Description	Default	Access
07	EN	**ACPI Enable:** 0 = Disable. Decoding of the I/O range pointed to by the ACPI base register is disabled, and the ACPI power management function is enabled. 1 = Decoding of the I/O range pointed to by the ACPI base register is enabled, and the ACPI power management function is enabled. ***Note:*** The APM power management ranges (B2/B3h) are always enabled and are not affected by this bit.	0	RW
06:03	Reserved	Reserved	0000	
02:00	SCIS	**SCI IRQ Select:** Specifies on which IRQ the SCI will internally appear. If not using the APIC, the SCI must be routed to IRQ9-11, and that interrupt is not sharable with the SERIRQ stream, but is shareable with other PCI interrupts. If using the APIC, the SCI can also be mapped to IRQ20-23, and can be shared with other interrupts. Bits SCI Map 000 IRQ9 001 IRQ10 010 IRQ11 011 Reserved 100 RQ20 (only if APIC enabled) 101 IRQ21 (only if APIC enabled) 110 IRQ22 (only if APIC enabled) 111 IRQ23 (only if APIC enabled) ***Note:*** When the interrupt is mapped to APIC interrupts 9, 10 or 11, the APIC must be programmed for active-high reception. When the interrupt is mapped to APIC interrupts 20 through 23, the APIC must be programmed for active-low reception.	000	RW

16.2.2.3 Offset 48h: GBA – GPIO Base Address Register

Table 563. Offset 48h: GBA – GPIO Base Address Register

Device: 31	**Function:** 0
Offset: 48h	**Size:** *32 bit*
Default Value: *00000001h*	**Power Well:** Core

Bits	Name	Description	Default	Access
31:16	Reserved	Reserved.	0000h	
15:06	BA	**Base Address:** Provides the 64 bytes of I/O space for GPIO.	00h	RW
05:01	Reserved	Reserved.	00h	
00	HD	Hardwired to 1 to indicate I/O space.	1	RO

16.2.2.4 Offset 4Ch: GC – GPIO Control Register

Table 564. Offset 4Ch: GC – GPIO Control Register

Device: 31	***Function:*** 0
Offset: 4Ch	***Size:*** *8 bit*
Default Value: *00h*	***Power Well:*** Core

Bits	Name	Description	Default	Access
07:05	Reserved	Reserved.	000	
04	EN	**GPIO Enable:** This bit enables/disables decode of the I/O range pointed to by the GPIO Base Address register (D31, F0, 48h) and enables the GPIO function. 0 = Disable 1 = Enable	0	RW
03:00	Reserved	Reserved.	0000	

16.2.3 Interrupt Configuration Registers

16.2.3.1 Offset 60h: PARC – PIRQ[A,B,C,D,E,F,G,H] Routing Control Register

Table 565. Offset 60h: PARC – PIRQ[A,B,C,D,E,F,G,H] Routing Control Register

Device: 31	***Function:*** 0
Offset: PIRQA — 60h, PIRQB — 61h PIRQC — 62h, PIRQD — 63h PIRQE — 68h, PIRQF — 69h PIROG — 6Ah. PIROH — 6Bh	***Size:*** *8 bit*

Bits	Name	Description	Default	Access
07	REN	**Interrupt Routing Enable:** 0 = The corresponding PIRQ is routed to one of the legacy interrupts specified in bits[03:00]. 1 = The PIRQ is not routed to the 8259. ***Note:*** BIOS must program this bit to 0 during POST for any of the PIRQs that are being used. The value of this bit may subsequently be changed by the OS when setting up for I/O APIC interrupt delivery mode.	1	RW
06:04	Reserved	Reserved	000	
03:00	IR	IRQ Routing: Bits Mapping Bits Mapping 0000 Reserved 1000 Reserved 0001 Reserved 1001 IRQ9 0010 Reserved 1010I RQ10 0011 IRQ3 1011 IRQ11 0100 IRQ4 1100 IRQ12 0101 IRQ5 1101 Reserved 0110 IRQ6 1110 IRQ14 0111 IRQ7 1111 IRQ15	0h	RW

16.2.3.2 Offset 64h: SCNT – Serial IRQ Control Register

Table 566. Offset 64h: SCNT – Serial IRQ Control Register

Device: 31	***Function:*** 0
Offset: 64h	***Size:*** *8 bit*
Default Value: *10h*	***Power Well:*** Core

Bits	Name	Description	Default	Access
07	EN	Enable: 0 = Serial IRQs will not be recognized 1 = Serial IRQs are recognized	0	RW
06	MD	Mode: 0 = The serial IRQ machine is in quiet mode 1 = The serial IRQ machine is in continuous mode ***Note:*** For systems using Quiet Mode, this bit must be set to 1 (Continuous Mode) for at least one frame after coming out of reset before switching back to Quiet Mode. Failure to do so will result in the IICH not recognizing SERIRQ interrupts.	0	RW
05:02	FS	**Frame Size:** 100 = Hardwired to indicate the size of the SERIRQ frame is 21 frames.	0100b	RO
01:00	SFPW	**Start Frame Pulse Width:** This is the number of 33 MHz clocks that the SERIRQ pin is driven low by the Serial IRQ controller to signal a start frame. In continuous mode, the controller will drive the start frame for the number of clocks specified. In quiet mode, the controller will drive the start frame for the number of clocks specified minus one, as the first clock was driven by the peripheral. Bits Clocks 00 4 01 6 10 8 11 Reserved	00	RW

When exiting S3/S5, the following algorithm must be used if the system needs Quiet Mode. Set the SERIRQ logic to continuous mode for at least one frame before switching it back to quiet mode.

16.2.4 LPC I/O Configuration Registers

16.2.4.1 Offset 80 - 81h: IOD – I/O Decode Ranges Register

Table 567. Offset 80 - 81h: IOD – I/O Decode Ranges Register (Sheet 1 of 2)

Device: 31	***Function:*** 0
Offset: 80 - 81h	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Default	Access
15:13	Reserved	Reserved	000	
12	FDD	**FDD Range:** This field determines which range to decode for the FDD Port 0 = 3F0 – 3F5h, 3F7h (Primary) 1 = 370 – 375h, 377h (Secondary)	0	RW
11:10	Reserved	Reserved	000	

Table 567. Offset 80 - 81h: IOD – I/O Decode Ranges Register (Sheet 2 of 2)

Device: 31	**Function:** 0
Offset: 80 - 81h	**Size:** 16 bit
Default Value: 0000h	**Power Well:** Core

Bits	Name	Description	Default	Access
09:08	LPT	**LPT Range:** This field determines which range to decode for the LPT Port: 00 378 – 37Fh and 778 – 77Fh 01 278 – 27Fh (port 279h is read only) and 678 – 67Fh 10 3BC –3BEh and 7BC – 7BEh 11 Reserved	0	RW
07	Reserved	Reserved	0	
06:04	CB	**ComB Range:** This field determines which range to decode for the COMB Port. 000 3F8 – 3FFh (COM1) 001 2F8 – 2FFh (COM2) 010 220 – 227h 011 228 – 22Fh 100 238 – 23Fh 101 2E8 – 2EFh (COM4) 110 338 – 33Fh 111 3E8 – 3EFh (COM3)	000	RW
03	Reserved	Reserved	0	
02:00	CA	**ComA Range:** This field determines which range to decode for the COMA Port. 000 3F8 – 3FFh (COM1) 001 2F8 – 2FFh (COM2) 010 220 – 227h 011 228 – 22Fh 100 238 – 23Fh 101 2E8 – 2EFh (COM4) 110 338 – 33Fh 111 3E8 – 3EFh (COM3)	000	RW

16.2.4.2 Offset 82 - 83h: IOE – I/O Enables Register

Table 568. Offset 82 - 83h: IOE – I/O Enables Register

Device: 31	**Function:** 0
Offset: 82 - 83h	**Size:** 16 bit
Default Value: 0000h	**Power Well:** Core

Bits	Name	Description	Default	Access
15:14	Reserved	Reserved	00	
13	ME2	**Micro controller Enable 2:** 0 = Disable 1 = Enables the decoding of the I/O locations 4Eh and 4Fh to the LPC interface. This range is used for a microcontroller.	0	RW
12	SE	**Super I/O Enable:** 0 = Disable 1 = Enables the decoding of the I/O locations 2Eh and 2Fh to the LPC interface. This range is used for Super I/O devices.	0	RW
11	ME1	**Micro controller Enable 1:** 0 = Disable 1 = Enables the decoding of the I/O locations 62h and 66h to the LPC interface. This range is used for a microcontroller.	0	RW
10	KE	**Keyboard Enable:** 0 = Disable 1 = Enables the decoding of the I/O locations 60h and 64h to the LPC interface. This range is used for a microcontroller.	0	RW
09	HGE	**High Gameport Enable:** 0 = Disable 1 = Enables decoding of the I/O locations 208h to 20Fh to LPC	0	RW
08	LGE	**Low Gameport Enable:** 0 = Disable 1 = Enables decoding of the I/O locations 200h to 207h to LPC	0	RW
07:04	Reserved	Reserved	0h	
03	FDE	**Floppy Drive Enable:** 0 = Disable 1 = Enables decoding of the FDD range to LPC. Range is selected by LIOD.FDE Decode Range Register (D31, F0, 80h, bit 12)	0	RW
02	PPE	**Parallel Port Enable:** 0 = Disable 1 = Enables decoding of the LPT range to LPC. Range is selected by LIOD.LPT Decode Range Register (D31, F0, 80h, bit 09:08)	0	RW
01	CBE	**Com Port B Enable:** 0 = Disable 1 = Enables decoding of the COMB range to LPC. Range is selected LIOD.CB Decode Range Register (D31, F0, 80h, bits 06:04)	0	RW
00	CAE	**Com Port A Enable:** 0 = Disable 1 = Enables decoding of the COMA range to LPC. Range is selected LIOD.CA Decode Range Register (D31, F0, 80h, bits 02:00)	0	RW

16.2.4.3 Offset 84 - 85h: LG1 – LPC Generic Decode Range 1 Register

Table 569. Offset 84 - 85h: LG1 – LPC Generic Decode Range 1 Register

Device: 31	***Function:*** 0
Offset: 84 - 85h	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Default	Access
15:07	BA	**Base Address:** Base Address for this generic decode range. This address is aligned on a 128-byte boundary, and being I/O, must have address lines 31:16 as 0. ***Note:*** This generic decode is for I/O addresses only, not memory addresses. The size of this range is 128 bytes.	0	RW
06:01	Reserved	Reserved.	0	
00	EN	**Enable:** 0 = Disable 1 = Enables the range specified in BA to be forwarded to LPC Interface	0	RW

16.2.4.4 Offset 88h: LG2 – LPC Generic Decode Range 2 Register

Table 570. Offset 88h: LG2 – LPC Generic Decode Range 2 Register

Device: 31	***Function:*** 0
Offset: 88h	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Default	Access
15:04	BA	**Base Address:** This address is aligned on a 16-byte boundary, and must have address lines 31:16 as 0.	0	RW
03:01	Reserved	Reserved.	0	
00	EN	**Enable:** 0 = Disable 1 = Enables the range specified in BA to be forwarded to LPC Interface	0	RW

16.2.5 Power Management Configuration Registers

Offsets A0h – CFh are described in the power management section. Refer to Chapter 22.0, “Power Management”.

16.2.6 FWH Configuration Registers

16.2.6.1 Offset D0 - D3h: FS1 – FWH ID Select 1 Register

This register contains the IDSEL fields the LPC Bridge uses for memory cycles going to the FWH.

Table 571. Offset D0 - D3h: FS1 – FWH ID Select 1 Register

Device: 31	***Function:*** 0
Offset: D0 - D3h	***Size:*** *32 bit*
Default Value: *00112233h*	***Power Well:***

Bits	Name	Description	Default	Access
31:28	IF8	**F8-FF IDSEL:** Is used in FWH cycle for ranges enabled by FDE.EF8. Used for two 512 kB Firmware Hub memory ranges and one 128 kB memory range. This field is fixed at 0000. The IDSEL programmed in this field addresses the following memory ranges: FFF8 0000h – FFFF FFFFh FFB8 0000h – FFBF FFFFh 000E 0000h – 000F FFFFh	0h	RO
27:24	IF0	**F0-F7 IDSEL:** Is used in FWH cycle for range enabled by FDE.EF0. Used for two 512 kB Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFF0 0000h – FFF7 FFFFh FFB0 0000h – FFB7 FFFFh	0h	RW
23:20	IE8	**E8-EF IDSEL:** Is used in FWH cycle for range enabled by FDE.EE8. Used for two 512 kB Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFE8 0000h – FFEF FFFFh FFA8 0000h – FFAF FFFFh	1h	RW
19:16	IE0	**E0-E7 IDSEL:** Is used in FWH cycle for range enabled by FDE.EE0. Used for two 512 kB Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFE0 0000h – FFE7 FFFFh FFA0 0000h – FFA7 FFFFh	1h	RW
15:12	ID8	**D8-DF IDSEL:** Is used in FWH cycle for range enabled by FDE.ED8. Used for two 512 kB Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFD8 0000h – FFDF FFFFh FF98 0000h – FF9F FFFFh	2h	RW
11:08	ID0	**D0-D7 IDSEL:** Is used in FWH cycle for range enabled by FDE.ED0. Used for two 512 kB Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFD0 0000h – FFD7 FFFFh FF90 0000h – FF97 FFFFh	2h	RW
07:04	IC8	**C8-CF IDSEL:** Is used in FWH cycle for range enabled by FDE.EC8. Used for two 512 kB Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFC8 0000h – FFCF FFFFh FF88 0000h – FF8F FFFFh	3h	RW
03:00	IC0	**C0-C7 IDSEL:** Is used in FWH cycle for range enabled by FDE.EC0. Used for two 512 kB Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFC0 0000h – FFC7 FFFFh FF80 0000h – FF87 FFFFh	3h	RW

16.2.6.2 Offset D4 - D5h: FS2 – FWH ID Select 2 Register

This register contains the additional IDSEL fields the LPC Bridge uses for memory cycles going to the FWH.

Table 572. Offset D4 - D5h: FS2 – FWH ID Select 2 Register

Device: 31	**Function:** 0
Offset: D4 - D5h	**Size:** *16 bit*
Default Value: *4567h*	**Power Well:** Core

Bits	Name	Description	Default	Access
15:12	I70	**70-7F IDSEL:** Is used in FWH cycle for range enabled by FDE.E70. Used for two, 1-M Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FF70 0000h – FF7F FFFFh FF30 0000h – FF3F FFFFh	4h	RW
11:08	I60	**60-6F IDSEL:** Is used in FWH cycle for range enabled by FDE.E60. Used for two, 1-M Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FF60 0000h – FF6F FFFFh FF20 0000h – FF2F FFFFh	5h	RW
07:04	I50	**50-5F IDSEL:** Is used in FWH cycle for range enabled by FDE.E50. Used for two, 1-M Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FF50 0000h – FF5F FFFFh FF10 0000h – FF1F FFFFh	6h	RW
03:00	I40	**40-4F IDSEL:** Is used in FWH cycle for range enabled by FDE.E40. Used for two, 1-M Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FF40 0000h – FF4F FFFFh FF00 0000h – FF0F FFFFh	7h	RW

16.2.6.3 Offset D8 - DBh: FDE – FWH Decode Enable Register

Table 573. Offset D8 - DBh: FDE – FWH Decode Enable Register (Sheet 1 of 3)

Device: 31	**Function:** 0
Offset: D8 - DBh	**Size:** *16 bit*
Default Value: *FFCFh*	**Power Well:** Core

Bits	Name	Description	Default	Access
15	EF8	**F8-FF Enable:** Enables decoding of 512 Kbyte Firmware Hub memory ranges: 0 = Disable 1 = Enable the following ranges for the Firmware Hub Data space:FFF80000h – FFFFFFFFh Feature space:FFB80000h – FFBFFFFFh	1	RO
14	EF0	**F0-F8 Enable:** Enables decoding of 512K-Kbyte Firmware Hub memory ranges: 0 = Disable. 1 = Enable the following ranges for the Firmware Hub: Data space:FFF00000h – FFF7FFFFh Feature space:FFB00000h – FFB7FFFFh	1	RW
13	EE8	**E8-EF Enable:** Enables decoding of 512K-Kbyte Firmware Hub memory ranges: 0 = Disable. 1 = Enable the following ranges for the Firmware Hub: Data space:FFE80000h – FFEFFFFFh Feature space:FFA80000h – FFAFFFFFh	1	RW

Table 573. Offset D8 - DBh: FDE – FWH Decode Enable Register (Sheet 2 of 3)

Device: 31	**Function:** 0
Offset: D8 - DBh	**Size:** 16 bit
Default Value: FFCFh	**Power Well:** Core

Bits	Name	Description	Default	Access
12	EE0	**E0-E8 Enable:** Enables decoding of 512K-Kbyte Firmware Hub memory ranges: 0 = Disable. 1 = Enable the following ranges for the Firmware Hub: Data space:FFE00000h – FFE7FFFFh Feature Space:FFA00000h – FFA7FFFFh	1	RW
11	ED8	**D8-DF Enable:** Enables decoding of 512K-Kbyte Firmware Hub memory ranges: 0 = Disable. 1 = Enable the following ranges for the Firmware Hub: Data space:FFD80000h – FFDFFFFFh Feature space:FF980000h – FF9FFFFFh	1	RW
10	ED0	**D0-D7 Enable:** Enables decoding of 512K-Kbyte Firmware Hub memory ranges: 0 = Disable. 1 = Enable the following ranges for the Firmware Hub: Data space:FFD00000h – FFD7FFFFh Feature space:FF900000h – FF97FFFFh	1	RW
09	EC8	**C8-CF Enable:** Enables decoding of 512K-Kbyte Firmware Hub memory ranges: 0 = Disable. 1 = Enable the following ranges for the Firmware Hub: Data space:FFC80000h – FFCFFFFFh Feature space:FF880000h – FF8FFFFFh	1	RW
08	EC0	**C0-C7 Enable:** Enables decoding of 512K-Kbyte Firmware Hub memory ranges: 0 = Disable. 1 = Enable the following ranges for the Firmware Hub: Data space:FFC00000h – FFC7FFFFh Feature space:FF800000h – FF87FFFFh	1	RW
07	LFE	**Legacy F Segment Enable:** This enables the decoding of the legacy 128K range at F0000h – FFFFFh 0 = Disable. 1 = Enable the following legacy ranges for the Firmware Hub F0000h – EFFFFh	1	RW
06	LEE	**Legacy E Segment Enable:** This enables the decoding of the legacy 128K range at E0000h – EFFFFh 0 = Disable. 1 = Enable the following legacy ranges for the Firmware Hub E0000h – EFFFFh	1	RW
05:04	Reserved	Reserved	00	
03	E70	**70-7F Enable:** Enables decoding of 1 MByte Firmware Hub memory range: Data space:FF700000h – FF7FFFFFh Feature space:FF300000h – FF3FFFFFh	1	RW

Table 573. Offset D8 - DBh: FDE – FWH Decode Enable Register (Sheet 3 of 3)

Device: 31	**Function:** 0
Offset: D8 - DBh	**Size:** *16 bit*
Default Value: *FFCFh*	**Power Well:** Core

Bits	Name	Description	Default	Access
02	E60	**60-6F Enable:** Enables decoding of 1 MByte Firmware Hub memory range: Data space:FF600000h – FF6FFFFFh Feature Space:FF200000h – FF2FFFFFh	1	RW
01	E50	**50-5F Enable:** Enables decoding of 1 MByte Firmware Hub memory range: Data space:FF500000h – FF5FFFFFh Feature space:FF100000h – FF1FFFFFh	1	RW
00	E40	**40-4F Enable:** Enables decoding of 1 MByte Firmware Hub memory range: Data space:FF400000h – FF4FFFFFh Feature space:FF000000h – FF0FFFFFh	1	RW

16.2.6.4 Offset DCh: BC – BIOS Control Register

Table 574. Offset DCh: BC – BIOS Control Register

Device: 31	**Function:** 0
Offset: DCh	**Size:** *8 bit*
Default Value: *00h*	**Power Well:** Core

Bits	Name	Description	Default	Access
07:02	Reserved	Reserved	0	
01	LE	**Lock Enable:** 0 = Setting the WP will not cause SMIs 1 = Enables setting the WP bit to cause SMIs. Once set, this bit can only be cleared by a PLTRST#	0	RWLO
00	WP	**Write Protect:** 0 = Only read cycles result in Firmware Hub Interface cycles 1 = Access to the BIOS space is enabled for both read and write cycles. When this bit is written from a 0 to a 1 and Lock Enable (LE) is also set, an SMI# is generated. This ensures that only SMI code can update BIOS.	0	RW

16.2.7 Root Complex Register Block Configuration Register

16.2.7.1 Offset F0h: RCBA – Root Complex Base Address Register

Table 575. Offset F0h: RCBA – Root Complex Base Address Register

Device: 31	***Function:*** 0
Offset: F0h	***Size:*** *32 bit*
Default Value: *00000000h*	***Power Well:*** Core

Bits	Name	Description	Default	Access
31:14	BA	**Base Address:** Base Address for the root complex register block decode range. This address is aligned on a 16 Kbyte boundary.	0	RW
13:01	Reserved	Reserved	0	
00	EN	**Enable:** 0 = Disables the range specified in BA to be claimed as the RCRB (Root Complex Register Block) 1 = Enables the range specified in BA to be claimed as the RCRB	0	RW

16.2.8 Manufacturing Information Register

16.2.8.1 Offset F8h: MANID – Manufacturer's ID Register

Table 576. Offset F8h: MANID – Manufacturer's ID

Device: 31	***Function:*** 0
Offset: F8h	***Size:*** *32 bit*
Default Value: *00010F80h*	***Power Well:*** Core

Bits	Name	Description	Default	Access
31:24	Reserved	Reserved	00h	
23:16	SID	**Stepping Identifier:** This field increments for each stepping of the part. ***Note:*** This field can be used by software to differentiate steppings when the Revision ID may not change. A single Stepping ID can be implemented that is readable from all functions in the chip because all of them increment in lock-step.	01h = A1	RO
15:08	MID	**Manufacturing Identifier:** 0Fh = Intel	0Fh	RO
07:00	Reserved	Reserved.	80h	

16.3 Interface

The LPC bridge function resides in Device 31, Function 0. In addition to the LPC bridge function, D31, F0 contains other functional units including DMA, Interrupt controllers, Timers, Power Management, System Management, GPIO, and RTC.

16.3.1 Overview

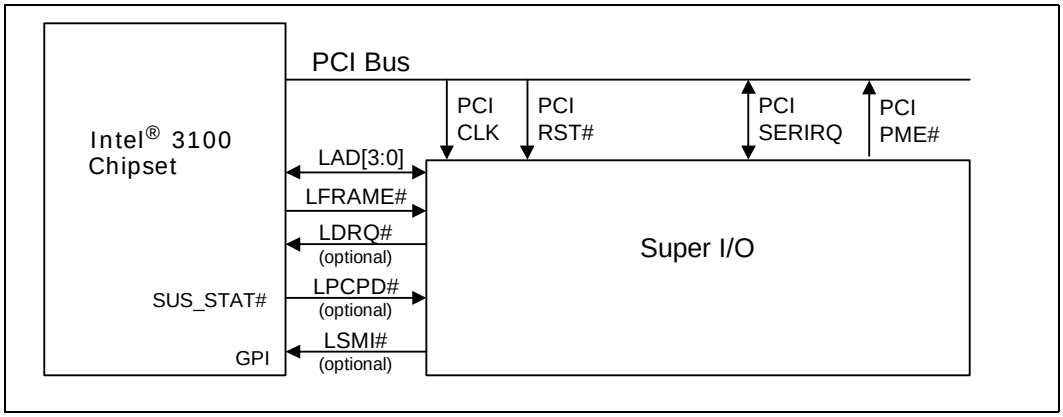
The LPC interface is described in the *Low Pin Count (LPC) Interface Specification, Revision 1.1*. The LPC interface to the IICH is shown in Figure 77. The LPC Controller implements all of the signals that are shown as optional, but peripherals are not required to do so.

For the LPC Controller:

- LSMI# can be connected to any of the SMI capable GPIO signals.
- The Super I/O's PME# can be connected to the PCI PME# signal, however this may cause software problems. A better choice is to connect it to one of the LPC Controller's SCI capable GPIO signals.
- The LPC Controller's SUS_STAT# signal is connected directly to the LPCPD# signal.

All the other signals have the same name on the LPC Controller and on the LPC Interface.

Figure 77. LPC Interface Diagram



16.3.2 Cycle Types

All of the cycle types implemented are described in the *LPC Interface Specification, Revision 1.1*. Table 577 shows the supported cycle types.

Table 577. LPC Cycle Types Supported

Cycle Type	Comment
Memory Read	Single: 1 byte only
Memory Write	Single: 1 byte only
I/O Read	1 byte only. breaks up 16- and 32-bit processor cycles into multiple 8-bit transfers. (See Note 1 below)
I/O Write	1 byte only. breaks up 16- and 32-bit processor cycles into multiple 8-bit transfers. (See Note 1 below)
DMA Read	Can be 1, or 2 bytes
DMA Write	Can be 1, or 2 bytes
Bus Master Read	Can be 1, 2, or 4 bytes. (See Note 2 below)
Bus Master Write	Can be 1, 2, or 4 bytes. (See Note 2 below)

Notes:

1. For memory cycles below 16 MB that do not target enabled firmware hub ranges, performs standard LPC memory cycles. It only attempts 8-bit transfers. If the cycle appears on PCI as a 16-bit transfer, it appears as two consecutive 8-bit transfers on LPC. Likewise, if the cycle appears as a 32-bit transfer on PCI, it appears as four consecutive 8-bit transfers on LPC. If the cycle is not claimed by any peripheral, it is subsequently aborted, and returns a value of all ones to the processor. This is done to maintain compatibility with legacy memory cycles where pull-up resistors would keep the bus high if no device responds.
2. Bus Master Read or Write cycles must be naturally aligned. For example, a 1-byte transfer can be to any address. However, the 2-byte transfer must be word aligned (i.e., with an address where A0=0). A DWord transfer must be DWord aligned (i.e., with an address where A1 and A0 are both 0).

The *LPC Specification* allows DMA cycles to be 4-bytes in length, but the LPC controller will only allow a maximum of 16-bit transfers. Additionally, the *LPC Specification* allows for firmware memory cycles to be 1, 2, or 4 bytes, and in the case of firmware reads, 128 bytes. However, the LPC controller will only perform 8-bit transfers.

Bus master read or write cycles must be naturally aligned. A 1 byte transfer can be to any address. A 2-byte transfer must be word aligned (address bit A0 = 0). A 4-byte transfer must be Dword aligned (address bits A[1:0] = 00).

16.3.3 Aborting a Cycle

The usage of LFRAME# is followed as it is defined in the *LPC Specification*.

The LPC Controller performs an abort for the following cases (possible failure cases):

- LPC Controller starts a Memory, I/O, or DMA cycle, but no device drives a valid SYNC after four consecutive clocks.
- LPC Controller starts a Memory, I/O, or DMA cycle, and the peripheral drives an invalid SYNC pattern.
- A peripheral drives an illegal address when performing bus master cycles.
- A peripheral drives an invalid value.

16.3.4 Memory Cycle Notes

For cycles below 16M and not targeting firmware, the LPC Controller will perform standard LPC memory cycles. For cycles targeting firmware, firmware memory cycles are used. For cycles targeting the fixed token, the fixed token format is used. Only 8-bit transfers are performed. If a larger transfer occurs, the LPC controller will break it into multiple 8-bit transfers until the request is satisfied.

Note: If the cycle is not claimed by any peripheral (and subsequently aborted), a value of all 1s (FFh) is returned to the processor. This is to maintain compatibility with legacy memory cycles where pull-up resistors would keep the bus high if no device responds.

16.3.5 I/O Cycle Notes

For I/O cycles targeting registers specified in the decode ranges, the Intel® 3100 Chipset performs I/O cycles as defined in the *LPC Specification*. These are 8-bit transfers. If the processor attempts a 16-bit or 32-bit transfer, the Intel® 3100 Chipset breaks the cycle up into multiple 8-bit transfers to consecutive I/O addresses until the request is satisfied.

Note: If the cycle is not claimed by any peripheral (and subsequently aborted), the Intel® 3100 Chipset returns a value of all 1s (FFh) to the processor. This is to maintain compatibility with legacy I/O cycles where pull-up resistors would keep the bus high if no device responds.

16.3.6 DMA Cycle Notes

Only 8-bit and 16-bit DMA transfers are supported. Peripherals must not attempt 32-bit transfers.

16.3.7 Bus Master Cycle Notes

The Intel® 3100 Chipset supports Bus Master cycles and requests (using LDRQ#) as defined in the *LPC Specification*. The Intel® 3100 Chipset has two LDRQ# inputs, and thus supports two separate bus master devices. It uses the associated START fields for Bus Master 0 (0010b) or Bus Master 1 (0011b).

Note: The Intel® 3100 Chipset does not support LPC Bus Masters performing I/O cycles. LPC Bus Masters must only perform memory read or memory write cycles and must only target main memory.

16.3.8 FWH Cycle Notes

A FWH device is not allowed to assert an Error SYNC. If the LPC controller receives any SYNC returned from the device other than short wait (0101), long wait (0110), or ready more (0000) when running a FWH cycle, indeterminate results will occur.

16.3.9 LPC PD# Protocol

The LPCPD# pin must follow the same timings as for SUS_STAT#. See Chapter 20.0, "Processor Interface," for details on the SUS_STAT# signal.

Upon driving SUS_STAT# low, LPC peripherals will drive LDRQ# low or tri-state it. The LPC Controller must shut the LDRQ# input buffers. After driving SUS_STAT# active, the LPC Controller drives LFRAME# low, and tri-states (or drive low) LAD[3:0].

Note: The *Low Pin Count Interface Specification, Revision 1.1* defines the LPCPD# protocol where there is at least 30 µs from LPCPD# assertion to LRST# assertion. This specification explicitly states that this protocol only applies to entry/exit of low power states which does not include asynchronous reset events. Intel® 3100 Chipset asserts both SUS_STAT# (connects to LPCPD#) and PLTRST# (connects to LRST#) at the same time when the core logic is reset (via CF9h, PWROK, or SYS_RESET#, etc.). This is not inconsistent with the LPC LPCPD# protocol.

16.3.10 Cycle Posting Policies

Three main policies are assumed.

1. I/O cycles and memory read cycles from the processor are not posted. Memory write Cycles from the processor are posted.
2. DMA cycles can be pipelined. For example, after reading data from memory, the LPC Controller can then release PHOLD# while it writes the data to the peripheral on the LPC Interface. This is because there are no processor/SMI#-based retry capabilities for DMA cycles. In the other direction, after reading data from a peripheral, PHOLD# can be released while the DMA controller writes data to main memory.
3. When bus masters read from main memory, the LPC Controller can treat this much like DMA, and release the memory and PCI buses while the data is being transferred to the bus master on the LPC Interface. When a bus master writes to main memory, the LPC Controller can use the LPC Interface while the data is being written to main memory.

16.3.11 Configuration

16.3.11.1 LPC Interface Decoders

To allow the I/O cycles and memory mapped cycles to go to the LPC interface, the Intel® 3100 Chipset includes several decoders. During configuration, the Intel® 3100 Chipset must be programmed with the same decode ranges as the peripheral. The decoders are programmed via the Device 31, Function 0 configuration space at offset 80h–87h.

Note: Intel® 3100 Chipset cannot accept PCI write cycles from PCI-to-PCI bridges or devices with similar characteristics (specifically those with a “Retry Read” feature which is enabled) to an LPC device if there is an outstanding LPC read cycle towards the same PCI device or bridge. These cycles are not part of normal system operation, but may be encountered as part of platform validation testing using custom test fixtures.

16.3.11.2 Bus Master Device Mapping and START Fields

Bus Masters must have a unique START field. In the case of the LPC Controller, which supports two bus masters, it will drive 0010 for the START field for grants to bus master 0 (requested via LDRQ[0]#) and 0011 for grants to bus master 1 (requested via LDRQ[1]#.).

16.3.11.3 Firmware Memory IDSEL fields

The LPC Controller uses a unique IDSEL field for each EPROM. The IDSEL used is determined through the programming of the FS1 and FS2 configuration registers.

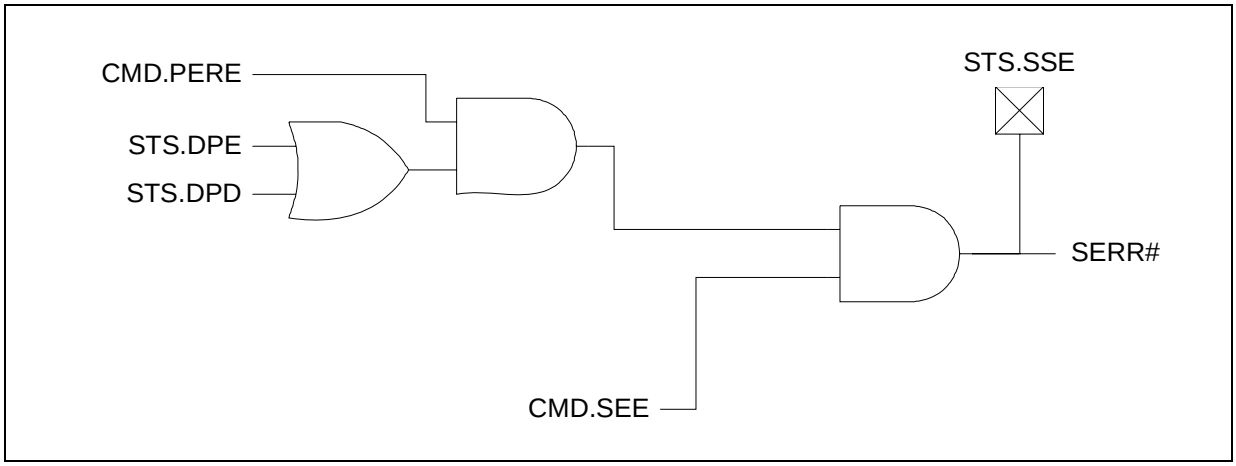
16.3.12 SERR# Generation

Several internal and external sources of the LPC Bridge can cause SERR#, and are described below.

The first class of errors is parity errors related internally to Intel® 3100 Chipset. The LPC Bridge captures generic data parity errors (errors it finds internally), as well as, errors returned on internal cycles where the bridge was the master. If either of these two conditions are met, and the bridge is enabled for parity error response, SERR# is captured.

Additionally, if the bridge receives a target abort or master abort, and the bridge policy is to SERR# on these types of aborts, SERR# is generated.

Figure 78. LPC Bridge SERR#



CMD.PERE - Offset 04-05h bit 06

STS.DPE - Offset 06-07h bit 15

STS.DPD - Offset 06-07h bit 08

STS.RTA - Offset 06-07h bit 12

STS.RMA - Offset 06-07h bit 13

17.0 LPC DMA

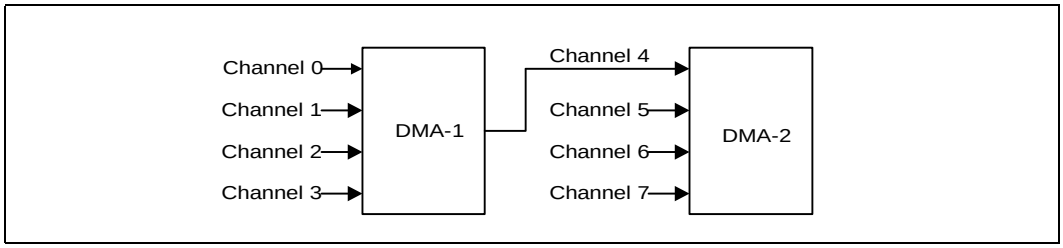
17.1 Overview

LPC DMA is supported using the IICH DMA controller. The DMA controller has registers that are fixed in the lower 64 Kbyte of I/O space. The DMA controller is configured using registers in the PCI configuration space. These registers allow configuration of the channels for use by LPC DMA.

The DMA circuitry incorporates the functionality of two 8237 DMA controllers with seven independently programmable channels; Channels 0–3 and Channels 5–7. DMA Channel 4 cascades the two controllers together and is not available for other use.

In addition to accepting requests from DMA slaves, the DMA controller also responds to requests that software initiates. Software may initiate a DMA service request by setting any bit in the DMA Channel Request Register to a one. The DMA controller is used for LPC DMA.

Figure 79. IICH DMA Controller



Feature set of the IICH DMA controller:

- Channels 0–3 provide 8-bit, count-by-bytes transfers.
- Channels 5–7 provide 16-bit, count-by-words transfers.
- 24-bit addressing. Each channel includes a 16-bit, legacy-compatible Current Address Register (CAR), which holds the 16 least-significant bits, and an legacy Compatible Page Register, which contains the eight next most significant bits of address.
- Auto-initialization following a DMA termination.

The DMA controller has registers that are fixed in the lower 64 Kbyte of I/O space.

The DMA controller is configured using registers in the PCI configuration space. These registers allow configuration of individual channels for use by LPC.

The RPR bit effects the register decoding for the Reserved Page register (Addresses 80, 84-86, 88, 8C-8E and their aliases in the 9x range). See Section 14.1.7.3, “Offset 3410 - 3413h: GCS – General Control and Status Register” bit 2.

The following decoding rules apply:

Rule 1: The IICH is performing subtractive decode

Rule 1.1: PCI READ and PCI WRITE cycles to Reserved page registers are decoded.

Rule 1.2: PCI WRITE cycles to Reserved page registers are NOT forwarded to LPC (See exception due to GCS.RPR.) The value is stored in the corresponding Reserved page register.

Rule 1.3: PCI READ cycles to Reserved page registers are NOT forwarded to LPC. The value returned is the value that is in the Reserved Page register.

17.2 DMA I/O Registers

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 578. DMA Registers Summary (Sheet 1 of 2)

Port	Alias	Symbol	Register Name/Function	Default	Access[1]
00h	10h		Channel 0 DMA Base and Current Address Register	XXXX	RW
01h	11h		Channel 0 DMA Base and Current Count Register	XXXXh	RW
02h	12h		Channel 1 DMA Base and Current Address Register	XXXX	RW
03h	13h		Channel 1 DMA Base and Current Count Register	XXXXh	RW
04h	14h		Channel 2 DMA Base and Current Address Register	XXXX	RW
05h	15h		Channel 2 DMA Base and Current Count Register	XXXXh	RW
06h	16h		Channel 3 DMA Base and Current Address Register	XXXX	RW
07h	17h		Channel 3 DMA Base and Current Count Register	XXXXh	RW
08h	18h		Channel 0-3 DMA Command Register	000X0X00h	WO
08h	18h		Channel 0-3 DMA Status Register	XXXXXXXh	RO
0Ah	1Ah		Channel 0-3 DMA Write Single Mask Register	000001XXh	WO
0Bh	1Bh		Channel 0-3 DMA Channel Mode Register	000000XXh	WO
0Ch	1Ch		Channel 0-3 DMA Clear Byte Pointer Register	XXXXXXXXh	WO
0Dh	1Dh		Channel 0-3 DMA Master Clear Register	XXXXXXXXh	WO
0Eh	1Eh		Channel 0-3 DMA Clear Mask Register	XXXXXXXXh	WO
0Fh	1Fh		Channel 0-3 DMA Write All Mask Register	00001111b	RW
80h	90h[1]		Reserved Page Register	Undefined	RW
81h	91h[1]		Channel 2 DMA Memory Low Page Register	XXXXXXXh	RW
82h	82h		Channel 3 DMA Memory Low Page Register	XXXXXXXh	RW
83h	93h[1]		Channel 1 DMA Memory Low Page Register	XXXXXXXh	RW
87h	97h[1]		Channel 0 DMA Memory Low Page Register	XXXXXXXh	RW
89h	99h[1]		Channel 6 DMA Memory Low Page Register	XXXXXXXh	RW
8Ah	9Ah[1]		Channel 7 DMA Memory Low Page Register	XXXXXXXh	RW
8Bh	9Bh[1]		Channel 5 DMA Memory Low Page Register	XXXXXXXh	RW
C4h	C5h		Channel 5 DMA Base and Current Address Register	XXXX	RW
C6h	C7h		Channel 5 DMA Base and Current Count Register	XXXXh	RW
C8h	C9h		Channel 6 DMA Base and Current Address Register	XXXX	RW

Notes:
1. Some registers are normally read-only, but are writable in Alt-Access mode. Likewise, there are some registers that are normally write-only, but are readable in Alt-Access mode. The individual register descriptions may not indicate this. See Alt-Access mode for more details.
2. The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 578. DMA Registers Summary (Sheet 2 of 2)

Port	Alias	Symbol	Register Name/Function	Default	Access[1]
CAh	CBh		Channel 6 DMA Base and Current Count Register	XXXXh	RW
CCh	CDh		Channel 7 DMA Base and Current Address Register	XXXX	RW
CEh	CFh		Channel 7 DMA Base and Current Count Register	XXXXh	RW
D0h	D1h		Channel 4-7 DMA Command Register	000X0X00h	WO
D0h	D1h		Channel 4-7 DMA Status Register	XXXXXXXh	RO
D4h	D5h		Channel 4-7 DMA Write Single Mask Register	000001XXb	WO
D6h	D7h		Channel 4-7 DMA Channel Mode Register	000000XXb	WO
D8h	D9h		Channel 4-7 DMA Clear Byte Pointer Register	XXXXXXXXh	WO
DAh	DBh		Channel 4-7 DMA Master Clear Register	XXXXXXXXh	WO
DCh	DDh		Channel 4-7 DMA Clear Mask Register	XXXXXXXXh	WO
DEh	DFh		Channel 4-7 DMA Write All Mask Register	00001111b	RW

Notes:

1. Some registers are normally read-only, but are writable in Alt-Access mode. Likewise, there are some registers that are normally write-only, but are readable in Alt-Access mode. The individual register descriptions may not indicate this. See Alt-Access mode for more details.
2. The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

17.2.1 Register Descriptions

17.2.1.1 DMA Base and Current Address Registers

Table 579. DMA Base and Current Address Registers

I/O Address: Ch. 0: 00h - 10h,
Ch. 1: 02h - 12h,
Ch. 2: 04h - 14h,
Ch. 3: 06h - 16h,
Ch. 5: C4h - C5h,
Ch. 6: C8h - C9h,
Ch. 7: CCh - CDh

Size: 16 bit (per channel), but accessed in two 8-bit transfers

Default Value: XXXX

Power Well: Core

Bits	Name	Description	Reset Value	Access
15:00	BCADD	**Base and Current Address:** This register determines the address for the transfers to be performed. The address specified points to two separate registers. On writes, the value is stored in the Base Address register and copied to the Current Address register. On reads, the value is returned from the Current Address register. The address increments/decrements in the Current Address register after each transfer, depending on the mode of the transfer. If the channel is in auto-initialize mode, the Current Address register is reloaded from the Base Address register after a terminal count is generated. For transfers to/from a 16-bit slave (channels 5-7), the address is shifted left one bit location. Bit 15 is shifted into bit 16. The register is accessed in 8-bit quantities. The byte is pointed to by the current byte pointer flip/flop. Before accessing an address register, the byte pointer flip/flop must be cleared to ensure that the low byte is accessed first.	XXXX	RW

17.2.1.2 DMA Base and Current Count Registers

Table 580. DMA Base and Current Count Registers

I/O Address:	Ch. 0: 01h - 11h, Ch. 1: 03h - 13h, Ch. 2: 05h - 15h, Ch. 3: 07h - 17h, Ch. 5: C6h - C7h, Ch. 6: CAh - CBh, Ch. 7: CEh - CFh	***Size:***	16 bit (per channel), but accessed in two 8-bit quantities
Default Value:	XXXXh	***Power Well:***	Core

Bits	Name	Description	Reset Value	Access
15:00	BCCNT	**Base and Current Count:** This register determines the number of transfers to be performed. The address specified points to two separate registers. On writes, the value is stored in the Base Count register and copied to the Current Count register. On reads, the value is returned from the Current Count register. The actual number of transfers is one more than the number programmed in the Base Count Register (i.e., programming a count of 4h results in 5 transfers). The count is decremented in the Current Count register after each transfer. When the value in the register rolls from zero to FFFFh, a terminal count is generated. If the channel is in auto-initialize mode, the Current Count register is reloaded from the Base Count register after a terminal count is generated. For transfers to/from an 8-bit slave (channels 0-3), the count register indicates the number of bytes to be transferred. For transfers to/from a 16-bit slave (channels 5-7), the count register indicates the number of words to be transferred. The register is accessed in 8-bit quantities. The byte is pointed to by the current byte pointer flip/flop. Before accessing a count register, the byte pointer flip/flop must be cleared to ensure that the lowest byte is accessed first.	XXXXh	RW

17.2.1.3 DMA Command Register

Table 581. DMA Command Register

I/O Address:	Ch. 0-3: 08h - 18h, Ch. 4-7: D0h - D1h	***Size:***	8 bit
Default Value:	000X0X00h	***Power Well:***	Core

Bits	Name	Description	Reset Value	Access
07:05	Reserved	Reserved. Software must always write 0 to these bits.	0	
04	DMAARB	**DMA Group Arbitration Priority:** Each channel group is individually assigned either fixed or rotating arbitration priority. At reset, each group is initialized in fixed priority. 0 = Assigns fixed priority to the channel group. 1 = Assigns rotating priority to the channel group.	X	WO
03	Reserved	Reserved. Must be 0	0	
02	DMACGEN	**DMA Channel Group Enable:** Both channel groups are enabled following part reset. Disabling channel group 4-7 also disables channel group 0-3, which is cascaded through channel 4. 0 = Enables the DMA channel group 1 = Disables the DMA channel group	X	WO
01:00	Reserved	Reserved. Must be 0.	0	

17.2.1.4 DMA Memory Low Page Registers

Table 582. DMA Memory Low Page Registers

I/O Address:	Ch. 0: 87h - 97h, Ch. 1: 83h - 93h, Ch. 2: 81h - 91h, Ch. 3: 82h, Ch. 5: 8Bh - 9Bh, Ch. 6: 89h - 99h, Ch. 7: 8Ah - 9Ah	***Size:***	8 bit
Default Value:	XXXXXXXh	***Power Well:***	Core
Lockable:	No		

Bits	Name	Description	Reset Value	Access
07:00	DMALP	**DMA Low Page (ISA Address bits [23:16]):** This register works in conjunction with the DMA controller's Current Address Register to define the complete 24-bit address for the DMA channel. This register remains static throughout the DMA transfer. Bit 16 of this register is ignored when in 16 bit I/O count by words mode as it is replaced by the bit 15 shifted out from the current address register.	X	RW

17.2.1.5 DMA Status Register

Table 583. DMA Status Register

I/O Address:	Ch. 0-3: 08h - 18h, Ch. 4-7: D0h - D1h	***Size:***	8 bit
Default Value:	XXXXXXXh	***Power Well:***	Core

Bits	Name	Description	Reset Value	Access
07:04	CRS	**Channel Request Status:** When a valid DMA request is pending for a channel, the corresponding bit is set to 1. When a DMA request is not pending for a particular channel, the corresponding bit is set to 0. The source of the DREQ may be hardware or a software request. ***Note:*** Channel 4 is the cascade channel, so the request status of channel 4 is a logical OR of the request status for channels 0 through 3. 4 Channel 0 5 Channel 1 (5) 6 Channel 2 (6) 7 Channel 3 (7)	XXXX	RO
03:00	CTCS	**Channel Terminal Count Status:** When a channel reaches terminal count (TC), its status bit is set to 1. If TC has not been reached, the status bit is set to 0. Channel 4 is programmed for cascade, so the TC bit response for channel 4 is irrelevant: 0 Channel 0 1 Channel 1 (5) 2 Channel 2 (6) 3 Channel 3 (7)	XXXX	RO

17.2.1.6 DMA Write Single Mask Register

Table 584. DMA Write Single Mask Register

I/O Address: Ch. 0-3: 0Ah - 1Ah, Ch. 4-7: D4h - D5h			***Size:*** 8 bit	
Default Value: 000001XXh			***Power Well:*** Core	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:03	Reserved	Reserved. Must be 0.	00000b	
02	CMS	**Channel Mask Select:** 0 = DREQ is enabled for the selected channel 1 = DREQ is disabled for the selected channel The channel is selected through bits [1:0]. Therefore, only one channel can be masked / unmasked at a time.	1	WO
01:00	DMACS	**DMA Channel Select:** These bits select which DMA Channel Mode Register is programmed. 00 Channel 0 (4) 01 Channel 1 (5) 10 Channel 2 (6) 11 Channel 3 (7)	XX	WO

17.2.1.7 DMA Channel Mode Register

Refer to the Glossary for definitions of DMA demand mode, single mode and cascade mode.

Table 585. DMA Channel Mode Register (Sheet 1 of 2)

I/O Address: Ch. 0-3: 0Bh - 1Bh, Ch. 4-7: D6h - D7h			***Size:*** 8 bit	
Default Value: 000000XXh			***Power Well:*** Core	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:06	DMATM	**DMA Transfer Mode:** Each DMA channel can be programmed in one of four different modes: 00 Demand mode 01 Single mode 10 Reserved 11 Cascade mode	00	WO
05	ADDIDS	**Address Increment/Decrement Select:** This bit controls address increment/decrement during DMA transfers. 0 = Address increment is selected. 1 = Address decrement is selected. Address increment is the default after part reset or Master Clear.	0	WO

Table 585. DMA Channel Mode Register (Sheet 2 of 2)

I/O Address: Ch. 0-3: 0Bh - 1Bh, Ch. 4-7: D6h - D7h			***Size:*** 8 bit	
Default Value: 000000XXh			***Power Well:*** Core	
Bits	**Name**	**Description**	**Reset Value**	**Access**
04	AUTOEN	**Autoinitialize Enable:** 0 = The autoinitialize feature is disabled and DMA transfers terminate on a terminal count. A part reset or Master Clear disables autoinitialization. 1 = The DMA restores the Base Address and Count registers to the current registers following a terminal count (TC).	0	WO
03:02	DMATT	**DMA Transfer Type:** These bits represent the direction of the DMA transfer. When the channel is programmed for cascade mode, (bits[07:06] = "11") the transfer type is irrelevant. 00 Verify - No I/O or memory strobes are generated 01 Write - Data transfers from the I/O devices to memory 10 Read - Data transfers from memory to the I/O device 11 Illegal	00	WO
01:00	DMACSEL	**DMA Channel Select:** These bits select the DMA Channel Mode Register that are written by bits [07:02]. 00 Channel 0 (4) 01 Channel 1 (5) 10 Channel 2 (6) 11 Channel 3 (7)	XX	WO

17.2.1.8 DMA Clear Byte Pointer Register

Table 586. DMA Clear Byte Pointer Register

I/O Address: Ch. 0-3: 0Ch - 1Ch, Ch. 4-7: D8h - D9h			***Size:*** 8 bit	
Default Value: XXXXXXXXh			***Power Well:*** Core	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:00	CBP	**Clear Byte Pointer:** Command is enabled with a write to the I/O port address. Writing to this register initializes the byte pointer flip/flop to a known state. It clears the internal latch used to address the upper or lower byte of the 16-bit Address and Word Count Registers. The latch is also cleared by a part reset and by the Master Clear command. This command precedes the first access to a 16 bit DMA controller register. The first access to a 16-bit register accesses the least significant byte, and the second accesses the most significant byte.	X	WO

17.2.1.9 DMA Master Clear Register

Table 587. DMA Master Clear Register

I/O Address:	Ch. 0-3: 0Dh - 1Dh, Ch. 4-7: DAh - DBh	***Size:***	8 bit
Default Value:	XXXXXXXXh	***Power Well:***	Core

Bits	Name	Description	Reset Value	Access
07:00	MSTCL	**Master Clear:** Enabled with a write to the port. This has the same effect as the hardware Reset; Command, Status, Request, and Byte Pointer flip/flop registers are cleared and the Mask Register is set.	X	WO

17.2.1.10 DMA Clear Mask Register

Table 588. DMA Clear Mask Register

I/O Address:	Ch. 0-3: 0Eh -1Eh, Ch. 4-7: DCh - DDh	***Size:***	8 bit
Default Value:	XXXXXXXXh	***Power Well:***	Core

Bits	Name	Description	Reset Value	Access
07:00	CLMR	**Clear Mask Register:** Command enabled with a write to the port.	X	WO

17.2.1.11 DMA Write All Mask Register

Table 589. DMA Write All Mask Register

I/O Address:	Ch. 0-3: 0Fh - 1Fh Ch. 4-7: DEh - DFh	***Size:***	8 bit
Default Value:	00001111b	***Power Well:***	Core

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved. Must be 0.	0	
03:00	CMSKB	**Channel Mask Bits:** Setting the bit(s) to a 1 disables the corresponding DREQ(s). Setting the bit(s) to a 0 enables the corresponding DREQ(s). Bits [03:00] are set to 1 upon part reset or Master Clear. When read, bits [03:00] indicate the DMA channel [03:00] ([07:04]) mask status. 0 Channel 0 (4) 1 Channel 1 (5) 2 Channel 2 (6) 3 Channel 3 (7) This register permits all four channels to be simultaneously enabled/disabled instead of enabling/disabling each channel individually, as is the case with the Mask Register - Write Single Mask Bit. This register also has a read path to allow the status of the channel mask bits to be read. A channel's mask bit is automatically set to 1 when the Current Byte/Word Count Register reaches terminal count (unless the channel is in auto-initialization mode). Disabling channel 4 also disables channels 0–3 due to the cascade of channels 0–3 through channel 4.	1111	RW

17.3 DMA Channel Arbitration

The IICH DMA controller consists of two logical channel groups; channels 0–3 and channels 4–7. Each group may be in either fixed or rotate mode as described in detail below.

The mode of operation for each controller is determined by the DMA Command register; address 08h for channels 0–3 and address D0h for channels 4–7. Since channels 0–3 are cascaded onto channel 4, any request on channel 0–3 appears as a request on channel 4.

The DMA controller stops rotating when an NMI is pending.

In fixed mode, the lowest numbered channel in a channel group receives highest priority. Therefore, channel 0 is the highest priority device of channels 0–3, and channel 4 is the highest priority device of channels 4–7. When both channels are programmed in fixed mode, channel 0 has highest priority and channel 7 the lowest.

In rotating mode, the lowest numbered channel starts out with highest priority. When it is serviced, the next numbered channel receives highest priority and the previous channel receives lowest priority. For example, if channel 0 has highest priority and is requesting, it wins arbitration, then is the lowest priority channel until channels 1, 2, and 3 have been serviced.

Due to the nature of channel 0–3 being cascaded onto channel 4, rotating mode adds some peculiarities to the arbitration scheme. Table 590 lists arbitration winners assuming all channels ares requesting.

Table 590. DMA Channel Priority

Current	Both Fixed	Lower Fixed, Upper Rotating	Lower Rotating, Upper Fixed	Both Rotating
0	0, 1, 2, 3, 5, 6, 7	5, 6, 7, 0, 1, 2, 3	1, 2, 3, 0, 5, 6, 7	5, 6, 7, 1, 2, 3, 0
1	0, 1, 2, 3, 5, 6, 7	5, 6, 7, 0, 1, 2, 3	2, 3, 0, 1, 5, 6, 7	5, 6, 7, 2, 3, 0, 1
2	0, 1, 2, 3, 5, 6, 7	5, 6, 7, 0, 1, 2, 3	3, 0, 1, 2, 5, 6, 7	5, 6, 7, 3, 0, 1, 2
3	0, 1, 2, 3, 5, 6, 7	5, 6, 7, 0, 1, 2, 3	0, 1, 2, 3, 5, 6, 7	5, 6, 7, 0, 1, 2, 3
5	0, 1, 2, 3, 5, 6, 7	6, 7, 0, 1, 2, 3, 5	0, 1, 2, 3, 5, 6, 7	6, 7, 0, 1, 2, 3, 5
6	0, 1, 2, 3, 5, 6, 7	7, 0, 1, 2, 3, 5, 6	0, 1, 2, 3, 5, 6, 7	7, 0, 1, 2, 3, 5, 6
7	0, 1, 2, 3, 5, 6, 7	0, 1, 2, 3, 5, 6, 7	0, 1, 2, 3, 5, 6, 7	0, 1, 2, 3, 5, 6, 7

17.4 Special Cases in Address/Count

17.4.1 Address Overrun/Underrun

Whenever the DMA is operating, the addresses do not increment or decrement through the High Page and Low Page registers. Therefore, if a 24-bit address is 01FFFFh and increments, the next address is 010000h, not 020000h. Similarly, if a 24-bit address is 020000h and decrements, the next address is 02FFFFh, not 01FFFFh.

However, when the DMA is operating in 16-bit mode, the addresses do not increment or decrement through the High Page and Low Page registers but the page boundary is now 128 Kbyte. Therefore, if a 24-bit address is 01FFFEh and increments, the next address is 000000h, not 010000h. Similarly, if a 24-bit address is 020000h and decrements, the next address is 03FFFEh, not 02FFFEh.

17.4.2 16-Bit Channels

For 16-bit channels, the DMA controller addressing is different than for 8-bit channels. The DMA controller shifts the lower 16 bits of address left 1 bit and shifts in a '0', as shown in Table 591. The count register is also redefined to represent words instead of bytes.

Table 591. Address Shifting in 16-bit DMA Transfers

Register			Address on 8 bit channels (hex)	Address on 16 bit channels (hex)
Page	High Byte	Low Byte		
00	01	01	00.01.01	00.02.02
01	FE	85	01.FE.85	01.FD.0A
01	FF	FF	01.FF.FF	01.FF.FE
00	FE	85	00.FE.85	01.FD.0A
00	FF	FF	00.FF.FF	01.FF.FE

17.4.3 Autoinitialize

By programming a bit in the DMA Channel Mode Register, a channel may be set up as an autoinitialize channel. When a channel undergoes autoinitialization, the original values of the Current Page, Current Address and Current Byte/Word Count Registers are automatically restored from the Base Page, Address, and Byte/Word Count Registers of that channel following TC. The Base Registers are loaded simultaneously with the Current Registers by the microprocessor when the DMA channel is programmed and remain unchanged throughout the DMA service. The mask bit is not set when the channel is in autoinitialize. Following autoinitialize, the channel is ready to perform another DMA service, without processor intervention, as soon as a valid DREQ is detected.

17.4.4 Software Commands

There are three additional special software commands that the DMA controller can execute. The three software commands are:

- Clear Byte Pointer Flip-Flop
- Master Clear
- Clear Mask Register

They are independent of any specific bit pattern on the data bus.

17.5 Theory of Operation for LPC DMA

DMA on LPC is handled through the use of the LDRQ# lines from peripherals and special encodings on LAD[3:0] from the host. Single, Demand, Verify, and Increment modes are supported on the LPC interface. Channel 0–3 are 8 bit channels. Channel 5–7 are 16 bit channels. Channel 4 is reserved as a generic bus master request (see Section 17.5.1 on LPC bus masters).

17.5.1 Asserting DMA Requests

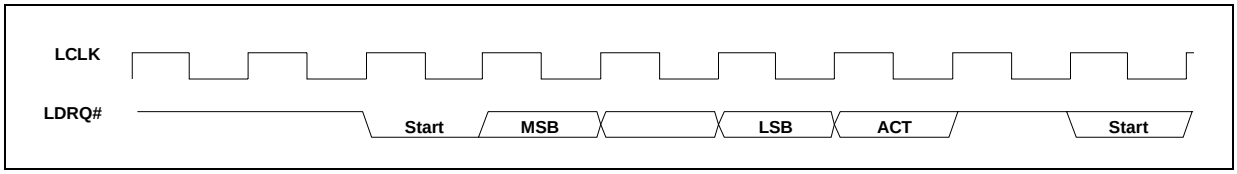
Peripherals that need DMA service encode their requested channel number on the LDRQ# signal. To simplify the protocol, each peripheral on the LPC Interface has its own dedicated LDRQ# signal (they may not be shared between two separate peripherals). The IICH has two LDRQ# inputs, allowing at least two devices to support DMA or bus mastering.

LDRQ# is synchronous with LCLK (PCI clock). As shown in Figure 80, the peripheral uses the following serial encoding sequence:

- Peripheral starts the sequence by asserting LDRQ# low (start bit). LDRQ# is high during idle conditions.
- The next three bits contain the encoded DMA channel number (MSB first).
- The next bit (ACT) indicates whether the request for the indicated DMA channel is active or inactive. The ACT bit is a 1 (high) to indicate if it is active and a 0 (low) if it is inactive. The case where ACT is low is rare, and is only used to indicate that a previous request for that channel is being abandoned. See Section 17.5.2 for reasons for abandoning DMA requests.
- After the active/inactive indication, the LDRQ# signal must go high for at least one clock. After that one clock, LDRQ# signal can be brought low to the next encoding sequence.

If another DMA channel also needs to request a transfer, another sequence can be sent on LDRQ#. For example, if an encoded request is sent for channel 2, and then channel 3 needs a transfer before the cycle for channel 2 is run on the interface, the peripheral can send the encoded request for channel 3. This allows multiple DMA agents behind an I/O device to request use of the LPC interface, and the I/O device does not need to self-arbitrate before sending the message.

Figure 80. DMA Request Assertion through LDRQ#



17.5.2 Abandoning DMA Requests

DMA Requests can be deasserted in two fashions: on error conditions by sending an LDRQ# message with the 'ACT' bit set to '0', or normally through a SYNC field during the DMA transfer. This section describes boundary conditions where the DMA request needs to be removed prior to a data transfer. See Section 17.5.6 to see how DMA requests are terminated through a DMA transfer.

There may be some special cases where the peripheral desires to abandon a DMA transfer. The most likely case of this occurring is due to a floppy disk controller which has overrun or enduring its FIFO, or software stopping a device prematurely.

In these cases, the peripheral wishes to stop further DMA activity. It may do so by sending an LDRQ# message with the ACT bit as '0'. However, since the DMA request was seen, there is no guarantee that the cycle hasn't been granted and runs on LPC. Therefore, peripherals must take into account that a DMA cycle may still occur. The peripheral can choose not to respond to this cycle, in which case the host aborts it, or it can choose to complete the cycle normally with any random data.

This method of DMA deassertion must be prevented whenever possible in order to limit boundary conditions both on the IICH and the peripheral.

The LDREQ DMA abort scheme should not be used if a transfer is in progress (a cycle has started) and more than one transfer has been completed. In these cases, the peripheral must use the SYNC field encoding 0000.

17.5.3 General Flow of DMA Transfers

Arbitration for DMA channels is performed through the 8237 within the host. Once the host has won arbitration on behalf of a DMA channel assigned to LPC, it asserts LFRAME# on the LPC Interface and begins the DMA transfer. The general flow for a basic DMA transfer is as follows:

1. The IICH starts the transfer by asserting ‘0000b’ on LAD[3:0] with LFRAME# asserted.
2. The IICH asserts ‘cycle type’ of DMA. The direction is based on the DMA transfer direction.
3. The IICH asserts the channel number and, if applicable, terminal count.
4. The IICH indicates the size of the transfer; 8 or 16 bits.
5. If a DMA reads:
   a. The IICH drives the first 8 bits of data and turns the bus around.
   b. The peripheral acknowledges the data with a valid SYNC.
   c. If a 16 bit transfer, the process is repeated for the next 8 bits.
6. If a DMA writes:
   a. The IICH turns the bus around and waits for data.
   b. The peripheral indicates data ready through SYNC and transfers the first byte.
   c. If a 16-bit transfer, the peripheral indicates data ready and transfers the next byte.
7. The peripheral turns around the bus.

17.5.4 Terminal Count

Terminal count is communicated through LAD[03] on the same clock that DMA channel is communicated on LAD[02:00]. This field is the CHANNEL field. Terminal count indicates the last byte of transfer, based upon the size of the transfer.

For example, on an 8-bit transfer size (SIZE field is ‘00b’), if the TC bit is set, then this is the last byte. On a 16-bit transfer (SIZE field is ‘01b’), if the TC bit is set, then the second byte is the last byte. The peripheral, therefore, must internalize the TC bit when the CHANNEL field is communicated and only signal TC when the last byte of that transfer size has been transferred.

17.5.5 Verify Mode

Verify mode is supported on the LPC interface. A verify transfer to the peripheral is similar to a DMA write, where the peripheral is transferring data to main memory. The indication from the host is the same as a DMA write, so the peripheral is driving data onto the LPC interface. However, the host does not transfer this data into main memory.

17.5.6 DMA Request Deassertion

An end of transfer is communicated to the IICH through a special SYNC field transmitted by the peripheral. If a DMA transfer is several bytes, such as a transfer from a demand mode device, the IICH needs to know when to deassert the DMA request based on the data currently being transferred.

The DMA agent uses a SYNC encoding on each byte of data being transferred, which indicates to the IICH whether this is the last byte of transfer or if more bytes are requested. To indicate the last byte of transfer, the peripheral uses a SYNC value of ‘0000b’ (ready with no error), or ‘1010b’ (ready with error). These encodings tell the IICH that this is the last piece of data transferred on a DMA read (IICH to peripheral), or the byte which follows is the last piece of data transferred on a DMA write (peripheral to IICH).

When the IICH sees one of these two encodings, it ends the DMA transfer after this byte and deasserts the DMA request to the 8237. Therefore, if the IICH indicated a 16 bit transfer, the peripheral can end the transfer after one byte by indicating a SYNC value of ‘0000b’ or ‘1010b’. The IICH does not attempt to transfer the second byte, and deasserts the DMA request internally. This allows the peripheral, therefore, to terminate a DMA burst.

If the peripheral indicates a ‘0000b’ or ‘1010b’ SYNC pattern on the last byte of the indicated size, then the IICH only deasserts the DMA request to the 8237 since it does not need to end the transfer.

If the peripheral wishes to keep the DMA request active, then it uses a SYNC value of ‘1001b’ (ready plus more data). This tells the 8237 that more data bytes are requested after the current byte has been transferred, so the IICH keeps the DMA request active to the 8237. Therefore, on an 8 bit transfer size, if the peripheral indicates a SYNC value of ‘1001b’, the data is transferred and the DMA request remains active to the 8237. At a later time, the IICH starts with another START ? CYCTYPE ? CHANNEL ? SIZE etc. combination to initiate another transfer to the peripheral.

The peripheral must not assume that the next START indication from the IICH is another grant to the peripheral if it had indicated a SYNC value of ‘1001b’. On a single mode DMA device, the 8237 rearbitrates after every transfer. Only demand mode DMA devices can be guaranteed that they receive the next START indication from the IICH.

Note: Indicating a ‘0000b’ or ‘1010b’ encoding on the SYNC field of an odd byte of a 16 bit channel (first byte of a 16 bit transfer) is an error condition.

Note: The host stops the transfer on the LPC bus as indicated, fill the upper byte with random data on DMA writes (peripheral to memory), and indicate to the 8237 that the DMA transfer occurred, incrementing the 8237s address and decrementing its byte count.

17.5.7 SYNC Field/LDRQ# Rules

Since DMA transfers on LPC are requested through an LDRQ# assertion message, and end through a SYNC field during the DMA transfer, the peripheral must obey the following rule when initiating back-to-back transfers from a DMA channel.

The peripheral must not assert another message for eight LCLKs after a deassertion is indicated through the SYNC field. This is needed to allow the 8237, which typically runs off a much slower internal clock, to see a message deasserted before it is reasserted so that it can arbitrate to the next agent.

Under default operation, the host only performs 8-bit transfers on 8-bit channels and 16-bit transfers on 16-bit channels. In order to enable 16-bit transfers on 8-bit channels, the peripheral must communicate to system BIOS that larger transfer sizes are allowed. If the host has this capability, the BIOS programs the host to attempt larger transfer sizes. IICH does not support 32-bit DMA transfer.

The method by which this communication between host and peripheral through system BIOS is performed is beyond the scope of this specification. Since the host and peripheral are motherboard devices, no “plug-n-play” registry is required.

The peripheral must not assume that the host is able to perform transfer sizes that are larger than the size allowed for the DMA channel, and be willing to accept a SIZE field that is smaller than what it may currently have buffered.

To that end, it is recommended that future devices which may appear on the LPC bus, which require higher bandwidth than 8-bit or 16-bit DMA allow, do so with a bus mastering interface and not rely on the 8237.

18.0 8254 Timers

18.1 Overview

The Intel® 3100 Chipset contains three counters which have fixed uses. All registers and functions associated with the 8254 timers are in the core power well. The 8254 unit is clocked by a 14.31818 MHz clock.

There is one signal associated with the 8254. It is used to drive the PC speaker.

Table 592. SPKR Signal

Signal Name	S3	S5
SPKR	Off	Off

18.2 8254 Timer Register Summary Table

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 593. 8254 Timer Register Summary Table

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
40h	40h		Counter 0 Interval Time Status Byte Format Register	0XXXXXXXb	RO
40h	40h		Counter 0 Counter Access Ports Register	XXh	RW
41h	41h		Counter 1 Interval Time Status Byte Format Register	0XXXXXXXb	RO
41h	41h		Counter 1 Counter Access Ports Register	XXh	RW
42h	42h		Counter 2 Interval Time Status Byte Format Register	0XXXXXXXb	RO
42h	42h		Counter 2 Counter Access Ports Register	XXh	RW
43h	43h	TCW	Timer Control Word Register	XXXXXXXh	WO, RWS

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

18.2.1 Counter 0, System Timer

This counter functions as the system timer by controlling the state of IRQ0 and is typically programmed for Mode 3 operation (described in Table 594). The counter produces a square wave with a period equal to the product of the counter period (838 ns) and the initial count value. The counter loads the initial count value one counter period after software writes the count value to the counter I/O address. The counter initially asserts IRQ0 and decrements the count value by two each counter period. The counter negates IRQ0 when the count value reaches 0. It then reloads the initial count

value and again decrements the initial count value by two each counter period. The counter then asserts IRQ0 when the count value reaches 0, reloads the initial count value and repeats the cycle, alternately asserting and negating IRQ0.

18.2.2 Counter 1, Refresh Request Signal

This counter provides the refresh request signal and is typically programmed for Mode 2 operation (described in Table 594) and only impacts the period of the REF_TOGGLE bit in Port 61. The initial count value is loaded one counter period after being written to the counter I/O address. The REF_TOGGLE bit has square wave behavior (alternate between 0 and 1) and toggles at a rate based on the value in the counter. Always program the counter to Mode 2. Programming other modes will result in undefined behavior for the REF_TOGGLE bit.

18.2.3 Counter 2, Speaker Tone

This counter provides the speaker tone and is typically programmed for Mode 3 operation. The counter provides a speaker frequency equal to the counter clock frequency (1.193 MHz) divided by the initial count value. The speaker must be enabled by a write to port 061h (See Section 20.1.1, “Register Descriptions” on page 649 for details).

18.2.4 Counter Operating Modes

Table 594 lists the six operating modes for the interval counters.

Table 594. Counter Operating Modes

Mode	Function	Description
0	Out signal on end of count (=0)	Output is ‘0’. When count goes to 0, output goes to ‘1’ and stays at ‘1’ until counter is reprogrammed.
1	Hardware retriggerable one-shot	Output is ‘0’. When count goes to 0, output goes to ‘1’ for one clock time.
2	Rate generator (divide by n counter)	Output is ‘1’. Output goes to ‘0’ for one clock time, then back to ‘1’ and counter is reloaded.
3	Square wave output	Output is ‘1’. Output goes to ‘0’ when counter rolls over, and counter is reloaded. Output goes to ‘1’ when counter rolls over, and counter is reloaded, etc.
4	Software triggered strobe	Output is ‘1’. Output goes to ‘0’ when count expires for one clock time.
5	Hardware triggered strobe	Output is ‘1’. Output goes to ‘0’ when count expires for one clock time.

18.3 Timer I/O Registers (LPC I/F – D31, F0)

18.3.1 Register Details

18.3.1.1 Offset 43h: TCW – Timer Control Word Register

This register is programmed prior to any counter being accessed to specify counter modes. The Interval Timer Status Byte Format Register can be found in Table 598 and the Counter Access Ports Register can be found in Table 599. Following reset, the control words for each register are undefined and each counter output is 0. Each timer must be programmed to bring it into a known state.

Table 595. Offset 43h: TCW – Timer Control Word Register

I/O Address: 43h	***Size:*** 8 bit
Default Value: XXXXXXXh	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:06	CNTSEL	**Counter Select:** The Counter Selection bits select the counter the control word acts upon as shown below. The Read Back command is selected when bits[07:06] are both one. 00 Counter 0 select 01 Counter 1 select 10 Counter 2 select 11 Read Back command	XXb	WO
05:04	RWMD	**Read/Write Mode Selection:** These bits are the read/write control bits. The actual counter programming is done through the counter port Table 599 (40h for counter 0, 41h for counter 1, and 42h for counter 2) 00 Counter Latch Command 01 Read/Write Least Significant Byte (LSB) 10 Read/Write Most Significant Byte (MSB) 11 Read/Write LSB then MSB	XXb	RWS
03:01	CNTMD	**Counter Mode Selection:** These bits select one of six possible modes of operation for the selected counter. **Reset Value / Mode / Mode Description** **000 / 0 / Out signal on end of count (=0)** **001 / 1 / Hardware retriggerable one-shot** **x10 / 2 / Rate generator (divide by n counter)** **x11 / 3 / Square wave output** **100 / 4 / Software triggered strobe** **101 / 5 / Hardware triggered strobe**	XXXb	WO
00	BCDCNT	**Binary/BCD Countdown Select:** 0 = Binary countdown is used. The largest possible binary count is 2^{16}. 1 = Binary coded decimal (BCD) count is used. The largest possible BCD count is 10^4.	Xb	WO

There are two special commands that can be issued to the counters through this register, the Counter Latch command (Section 18.5.2) and the Read Back command (Section 18.5.3). When these commands are chosen, several bits within this register are redefined. These register formats are described in Section 18.5.

18.4 Timer Programming

The counter/timers are programmed as follows:

1. Write a control word to select a counter.
2. Write an initial count for that counter.
3. Load the least and/or most significant bytes (as required by TCW.RWMD) of the 16-bit counter.
4. Repeat with other counters.

Only two conventions need to be observed when programming the counters. First, for each counter, the control word must be written before the initial count is written. Second, the initial count must follow the count format specified in the control word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

A new initial count may be written to a counter at any time without affecting the counter's programmed mode. Counting is affected as described in the mode definitions. The new count must follow the programmed count format.

If a counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same counter. Otherwise, the counter is loaded with an incorrect count.

The Control Word Register at port 43h controls the operation of all three counters. Several commands are available:

- **Control Word Command:** Specifies which counter to read or write, the operating mode, and the count format (binary or BCD).
- **Counter Latch Command:** Latches the current count so that it can be read by the system. The countdown process continues.
- **Read Back Command:** Reads the count value, programmed mode, the current state of the OUT pins, and the state of the Null Count Flag of the selected counter.

18.5 Reading from the Interval Timer

There are three methods for reading the counters: a Simple Read operation, Counter Latch command, and the Read-Back command. Each is explained in the following sections.

With the Simple Read and Counter Latch command methods, the count must be read according to the programmed format; specifically, if the counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read sequentially. Read, write, or programming operations for other counters may be inserted between the two byte reads.

None of these methods interfere with the count in progress.

18.5.1 Simple Read

The first method to read the counter is to perform a simple read operation. The counter is selected through port 40h (counter 0), 41h (counter 1), or 42h (counter 2).

Note: Performing a direct read from the counter does not return a determinate value, because the counting process is asynchronous to read operations. However, in the case of counter 2, the count can be stopped by writing to the GATE bit in port 61h.

18.5.2 Counter Latch Command

The Counter Latch Command (Table 596), written to port 43h, latches the count of the specific counter at the time the command is received. This command is used to ensure that the count read from the counter is accurate, particularly when reading a two-byte count. The count value is then read from each counter's Count Register through the Counter Access Ports Register Table 599 (40h for counter 0, 41h for counter 1, and 42h for counter 2).

The count is held in the latch until it is read or the counter is reprogrammed. The count is then unlatched. This allows reading the contents of the counters on the fly without affecting counting in progress. Multiple Counter Latch commands may be used to latch more than one counter. Counter Latch commands do not affect the programmed mode of the counter in any way.

If a counter is latched and then latched again before the count is read, the second Counter Latch command is ignored. The count read is the count at the time the first Counter Latch command was issued.

Table 596. Counter Latch Command

Bits	Description
07:06	**Counter Selection:** These bits select the counter for latching. 00 Counter 0 01 Counter 1 10 Counter 2 11 The write is interpreted as a read back command.
05:04	**Counter Latch Command:** Write '00' to select the Counter Latch Command.
03:00	Reserved. Must be 0.

18.5.3 Read Back Command

The Read Back command, written to port 43h, latches the count value, programmed mode, and current states of the OUT pin and Null Count flag of the selected counter or counters. The value of the counter and its status may then be read by I/O access to the counter address.

The Read Back command may be used to latch multiple counter outputs at one time. This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read or reprogrammed. Once read, a counter is unlatched. The other counters remain latched until they are read. If multiple count Read Back commands are issued to the same counter without reading the count, all but the first are ignored. Both count and status of the selected counters may be latched simultaneously by setting both bit 5 and bit 4 to 0. If both are latched, the first read operation from that counter returns the latched status. The next one or two reads, depending on whether the counter is programmed for one or two byte counts, returns the latched count. Subsequent reads return an unlatched count.

The Read Back command may additionally be used to latch status information of selected counters. The status of a counter is accessed by a read from that counter's I/O port address. If multiple counter status latch operations are performed without reading the status, all but the first are ignored.

Both count and status of the selected counters may be latched simultaneously in any or all of the counters by selecting the counter during the register write. This is functionally the same as issuing two consecutive, separate Read Back commands. If multiple count and/or status Read Back commands are issued to the same counters without any intervening reads, all but the first are ignored.

If both count and status of a counter are latched, the first read operation from that counter will return the latched status, regardless of which was latched first. The next one or two reads, depending on whether the counter is programmed for one or two type counts, return the latched count. Subsequent reads return unlatched count.

Table 597. Read Back Command

Bits	Description
07:06	**Read Back Command:** Must be '11' to select the Read Back Command.
05	**Latch Count of Selected Counters:** 0 = Current count value of the selected counters are latched 1 = Current count value of the selected counters are not latched
04	**Latch Status of Selected Counters:** 0 = Status of the selected counters are latched 1 = Status of the selected counters are not latched
03	**Counter 2 Select:** 0 = Counter 2 count and/or status are not latched 1 = Counter 2 count and/or status are latched
02	**Counter 1 Select:** 0 = Counter 1 count and/or status are latched 1 = Counter 1 count and/or status are not latched
01	**Counter 0 Select:** 0 = Counter 0 count and/or status are latched. 1 = Counter 0 count and/or status are not latched
00	Reserved. Must be 0.

18.5.3.1 Interval Timer Status Byte Format Register

Each counter's status byte can be read following a Read Back command. If latch status is chosen (bit 4=0, Read Back command) as a read back option for a given counter, the next read from the counter's Counter Access Ports Register (40h for counter 0, 41h for counter 1, and 42h for counter 2) returns the status byte. The status byte returns the values shown in Table 598.

Table 598. Interval Timer Status Byte Format Register

I/O Address: Counter 0: 40h, Counter 1: 41h, Counter 2: 42h			**Size:** 8 bit	
Default Value: 0XXXXXXXb			**Power Well:** Core	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07	COPS	**Counter OUT Pin State:** 0 = The OUT pin of the counter is a 0 1 = The OUT pin of the counter is a 1	0b	RO
06	CRS	**Count Register Status:** This bit indicates when the last count written to the Count Register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the counter mode, but until the count is loaded into the counting element (CE), the count value is incorrect. 0 = Count has been transferred from CR to CE and is available for reading. 1 = Null Count. Count has not been transferred from CR to CE and is not yet available for reading.	Xb	RO
05:04	RWSS	**Read/Write Selection Status:** These reflect the read/write selection made through bits[05:04] of the control register. The binary codes returned during the status read match the codes used to program the counter read/write selection. 00 Counter Latch Command 01 Read/Write Least Significant Byte (LSB) 10 Read/Write Most Significant Byte (MSB) 11 Read/Write LSB then MSB	XXb	RO
03:01	MSS	**Mode Selection Status:** These bits return the counter mode programming. The binary code returned matches the code used to program the counter mode, as listed under the bit function above. 000 0 Out signal on end of count (=0) 001 1 Hardware retriggerable one-shot x10 2 Rate generator (divide by n counter) x11 3 Square wave output 100 4 Software triggered strobe 101 5 Hardware triggered strobe	XXXb	RO
00	CTS	**Countdown Type Status:** This bit reflects the current countdown type; ether 0 for binary countdown or a 1 for binary coded decimal (BCD) countdown.	Xb	RO

18.5.3.2 Counter Access Ports Register

Table 599. Counter Access Ports Register

I/O Address: Counter 0: 40h, Counter 1: 41h, Counter 2: 42h			**Size:** 8 bit	
Default Value: XXh			**Power Well:** Core	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:00	CNTP	**Counter Port:** Each counter port address is used to program the 16-bit Count Register. The order of programming, either LSB only, MSB only, or LSB then MSB, is defined with the Interval Counter Control Register at port 43h. The counter port is also used to read the current count from the Count Register, and return the status of the counter programming following a Read Back command.	XXh	RW

19.0 Interrupts

19.1 Overview

Only level-triggered interrupts can be shared. PCI interrupts (PIRQs) are inherently shared on the board. These must, therefore, be programmed as level-triggered.

Table 600 and Table 601 show the mapping of the various interrupts in Non-APIC and APIC modes. Table 602 list the interrupt signals action in the associated power state

Table 600. Interrupt Options - 8259 Mode

IRQ	SERIRQ	Pin	Internal Modules
0	No	No	8254 Counter 0, MMT 0
1	Yes	No	
2	No	No	8259 2 cascade only
3	Yes	No	Option for PIRQx
4	Yes	No	Option for PIRQx
5	Yes	No	Option for PIRQx
6	Yes	No	Option for PIRQx
7	Yes	No	Option for PIRQx
8	No	No	RTC, MMT 1
9	Yes	No	Option for PIRQx, SCI, TCO
10	Yes	No	Option for PIRQx, SCI, TCO
11	Yes	No	Option for PIRQx, SCI, TCO, MMT 2
12	Yes	No	Option for PIRQx
13	No	No	FERR# Logic
14	Yes	Yes	PIRQx, S ATA Primary (legacy mode)
15	Yes	Yes	PIRQx, S ATA Secondary (legacy mode)

Notes:

1. If an interrupt is used for PCI IRQ[A:H], SCI, or TCO, it must not be used for ISA (legacy)-style interrupts (via SERIRQ).
2. In 8259 mode, PCI interrupts are mapped to IRQ3, 4, 5, 6, 7, 9, 10, 11, 12, 14, or 15.
3. If IRQ11 is used for MMT 2, software must ensure IRQ11 is not shared with any other devices to guarantee the proper operation of MMT 2. The hardware does not prevent sharing of IRQ11.

Table 601. Interrupt Options - APIC Mode

IRQ #	SERIRQ	Pin	PCI Message	Internal Modules
0	No	No	No	Cascade from 8259 1
1	Yes	No	Yes	
2	No	No	No	8254 Counter 0, MMT 0 (legacy mode)
3	Yes	No	Yes	
4	Yes	No	Yes	
5	Yes	No	Yes	
6	Yes	No	Yes	
7	Yes	No	Yes	
8	No	No	No	RTC, MMT 1 (legacy mode)
9	Yes	No	Yes	Option for SCI, TCO
10	Yes	No	Yes	Option for SCI, TCO
11	Yes	No	Yes	Option for SCI, TCO
12	Yes	No	Yes	
13	No	No	No	FERR# Logic
14	Yes	Yes	Yes	SATA Primary (legacy mode)
15	Yes	Yes	Yes	SATA Secondary (legacy mode)
16	PIRQA	PIRQA	Yes	See Section 19.5 for how internal devices are mapped.
17	PIRQB	PIRQB		
18	PIRQC	PIRQC		
19	PIRQD	PIRQD		
20	N/A	PIRQE		Option for SCI, TCO, and HPET (High Precision Event Timer). For other internal devices, see Section 19.5
21	N/A	PIRQF		
22	N/A	PIRQG		
23	N/A	PIRQH		

Notes:

1. If an interrupt is used for PCI IRQ[A:H], SCI, or TCO, it must not be used for ISA (legacy)-style interrupts (via SERIRQ).
2. In APIC mode, the PCI interrupts A:H are mapped to IRQ[16:23].
3. When programming the polarity of internal interrupt sources on the APIC, interrupts 0 through 15 receive active-high internal interrupt sources; interrupts 16 through 23 receive active-low internal interrupt sources.

Table 602. Signals Associated with Interrupt Logic

Signal Name	S3	S5
SERIRQ	Off	Off
PIRQ[A:H]#	Off	Off
IDEIRQ	Off	Off

19.2 8259 Interrupt Controllers (PIC)

19.2.1 Overview

The IICH incorporates the functionality of two 8259 interrupt controllers that provide system interrupts for the legacy (ISA) compatible interrupt controller (PIC). These interrupts are: system timer, keyboard controller, serial ports, parallel ports, floppy disk, mouse, and DMA channels. In addition, this interrupt controller can support the PCI based interrupts by mapping the PCI interrupt onto the compatible IRQ interrupt line. Each 8259 core supports eight interrupts, numbered 0–7. Table 603 shows how the cores are connected.

Table 603. 8259 Core Connection

8259	8259 Input	Typical Interrupt Source	Connected Pin / Function
Master	0	Internal	Internal Timer / Counter 0 output or Multimedia Timer 0
	1	Keyboard	IRQ1 via SERIRQ
	2	Internal	Slave Controller INTR output
	3	Serial Port A	IRQ3 via SERIRQ, PIRQx
	4	Serial Port B	IRQ4 via SERIRQ, PIRQx
	5	Parallel Port / Generic	IRQ5 via SERIRQ, PIRQx
	6	Floppy Disk	IRQ6 via SERIRQ, PIRQx
	7	Parallel Port / Generic	IRQ7 via SERIRQ, PIRQx
Slave	0	Real Time Clock	Inverted IRQ8# from internal RTC or Multimedia Timer 1
	1	Generic	IRQ9 via SERIRQ, SCI, or TCO, PIRQx
	2	Generic	IRQ10 via SERIRQ, SCI, or TCO, PIRQx
	3	Generic	IRQ11 via SERIRQ, SCI, or TCO, PIRQx
	4	PS/2 Mouse	IRQ12 via SERIRQ, SCI, or TCO, PIRQx
	5	Internal	State Machine output based on processor FERR# assertion. Can optionally be used for SCI or TCO interrupts if FERR# is not needed.
	6	SATA	SATA Primary (legacy mode), SERIRQ, PIRQx
	7	SATA	SATA Secondary (legacy mode), SATA Secondary (legacy mode), SERIRQ, PIRQx

The IICH cascades the slave controller onto the master controller through master controller interrupt input 2. This means there are only 15 possible interrupts for the IICH PIC.

Interrupts can individually be programmed to be edge or level, except for IRQ0, IRQ2, IRQ8#.

Note: Active-low interrupt sources, such as the PIRQ#s, are internally inverted in the IICH. In the following descriptions of the 8259s, the interrupt levels are in reference to the signals at the internal interface of the 8259s, after the required inversions have occurred. Therefore, the term “high” indicates “active”, which means “low” on an originating PIRQ#.

19.2.2 I/O Registers

The interrupt controller registers are located at 20h and 21h for the master controller (IRQ0–7), and at A0h and A1h for the slave controller (IRQ8–13). These registers have multiple functions depending upon the data written to them. Table 604 lists the different register possibilities for each address.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 604. 8259 Interrupt Controller (PIC) Registers (LPC I/F – D31, F0)

Port	Aliases	Register Name	Default Value	Type
20h	24h, 28h, 2Ch, 30h, 34h, 38h, 3Ch	Master PIC ICW1 Init. Cmd Word 1	Undefined	WO
		Master PIC OCW2 Op Ctrl Word 2	001XXXXXb	WO
		Master PIC OCW3 Op Ctrl Word 3	X01XXX10b	WO
21h	25h, 29h, 2Dh, 31h, 35h, 39h, 3Dh	Master PIC ICW2 Init. Cmd Word 2	Undefined	WO
		Master PIC ICW3 Init. Cmd Word 3	Undefined	WO
		Master PIC ICW4 Init. Cmd Word 4	01h	WO
		Master PIC OCW1 Op Ctrl Word 1	00h	RW
A0h	A4h, A8h, ACh, B0h, B4h, B8h, BCh	Slave PIC ICW1 Init. Cmd Word 1	Undefined	WO
		Slave PIC OCW2 Op Ctrl Word 2	001XXXXXb	WO
		Slave PIC OCW3 Op Ctrl Word 3	X01XXX10b	WO
A1h	A5h, A9h, ADh, B1h, B5h, B9h, BDh	Slave PIC ICW2 Init. Cmd Word 2	Undefined	WO
		Slave PIC ICW3 Init. Cmd Word 3	Undefined	WO
		Slave PIC ICW4 Init. Cmd Word 4	01h	WO
		Slave PIC OCW1 Op Ctrl Word 1	00h	RW
4D0h	–	Master PIC Edge/Level Triggered	00h	RW
4D1h	–	Slave PIC Edge/Level Triggered	00h	RW

Notes:
1. Refer to note addressing active-low interrupt sources in Section 19.2.1.
2. The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

19.2.2.1 ICW1 – Initialization Command Word 1

A write to Initialization Command Word 1 starts the interrupt controller initialization sequence, during which the following occurs:

1. The Interrupt Mask register is cleared.
2. IRQ7 input is assigned priority 7.
3. The slave mode address is set to 7.
4. Special Mask Mode is cleared and Status Read is set to IRR.

Once this write occurs, the controller expects writes to ICW2, ICW3, and ICW4 to complete the initialization sequence.

Table 605. ICW1 – Initialization Command Word 1

I/O Address: Master Controller - 020h
Slave Controller - 0A0h

Size: 8 bits per controller

Default Value: 0001X0XXb

Bits	Name	Description	Reset Value	Access
07:05	Reserved	Reserved. Must be programmed to zero.	000	
04	Reserved	Reserved. Must be programmed to one.	1	
03	LTIM	**Edge/Level Bank Select:** Disabled. Replaced by the edge/level triggered control registers (ELCR, D31, F0, 4D0h and D31, F0, 4D1h).	X	WO
02	Reserved	Reserved. Must be programmed to zero.	0	
01	SNGL	**Single or Cascade:** This bit must be programmed to a 0 to indicate that two controllers are operating in cascade mode.	X	WO
00	IC4	**ICW4 Write Required:** This bit must be programmed to a 1 to indicate that ICW4 needs to be programmed.	X	WO

19.2.2.2 Offset 21h/A1h: ICW2 – Initialization Command Word 2

ICW2 is used to initialize the interrupt controller with the five most significant bits of the interrupt vector address. The value programmed for bits[07:03] is used by the processor to define the base address in the interrupt vector table for the interrupt routines associated with each IRQ on the controller. Typical ISA (legacy) ICW2 values are 08h for the master controller and 70h for the slave controller.

Table 606. Offset 21h/A1h: ICW2 – Initialization Command Word 2

I/O Address: Master Controller - 021h
Slave Controller - 0A1h

Size: 8 bits per controller

Default Value: XXh

Bits	Name	Description	Reset Value	Access
07:03	IVBA	**Interrupt Vector Base Address:** These bits define the base address in the interrupt vector table for the interrupt routines associated with each interrupt request level input.	Xh	WO
02:00	IRL	**Interrupt Request Level:** When writing ICW2, these bits must all be 0. During an interrupt acknowledge cycle, these bits are programmed by the interrupt controller with the interrupt to be serviced. This is combined with bits [07:03] to form the interrupt vector driven onto the data bus during the second INTA# cycle. The code is a three bit binary code: Code Master Interrupt Slave Interrupt 000 IRQ0 IRQ8 001 IRQ1 IRQ9 010 IRQ2 IRQ10 011I RQ3 IRQ11 100 IRQ4 IRQ12 101 IRQ5 IRQ13 110 IRQ6 IRQ14 111 IRQ7 IRQ15	Xh	WO

19.2.2.3 Offset 21h: MICW3 – Master Initialization Command Word 3

Table 607. Offset 21h: MICW3 – Master Initialization Command Word 3

I/O Address: *21h* ***Size:*** 8 bit

Default Value: *04h*

Bits	Name	Description	Reset Value	Access
07:03	Reserved	Reserved. Must be programmed to zero.	0	
02	CCC	**Cascaded Controller Connection:** This bit must always be programmed to a 1. This bit indicates that the slave controller is cascaded on IRQ2. When IRQ8#–IRQ15 is asserted, it goes through the slave controller's priority resolver. The slave controller's INTR output onto IRQ2. IRQ2 then goes through the master controller's priority solver. If it wins, the INTR signal is asserted to the processor, and the returning interrupt acknowledge returns the interrupt vector for the slave controller.	1	WO
01:00	Reserved	Reserved. Must be programmed to zero.	0	

19.2.2.4 Offset A1h: SICW3 – Slave Initialization Command Word 3

Table 608. Offset A1h: SICW3 – Slave Initialization Command Word 3

I/O Address: A1h ***Size:*** 8 bit

Default Value: *00h*

Bits	Name	Description	Reset Value	Access
07:03	Reserved	Reserved. Must be programmed to zero.	0	
02:00	SIC	**Slave Identification Code:** These bits are compared against the slave identification code broadcast by the master controller from the trailing edge of the first internal INTA# pulse to the trailing edge of the second internal INTA# pulse. These bits must be programmed to 02h to match the code broadcast by the master controller. When 02h is broadcast by the master controller during the INTA# sequence, the slave controller assumes responsibility for broadcasting the interrupt vector.	0	WO

19.2.2.5 Offset 21h/A1h: ICW4 – Initialization Command Word 4 Register

Table 609. Offset 21h/A1h: ICW4 – Initialization Command Word 4 Register

I/O Address: Master Controller - 021h
Slave Controller - 0A1h

Size: 8 bits per controller

Default Value: *01h*

Bits	Name	Description	Reset Value	Access
07:05	Reserved	Must be programmed to zero.	0	
04	SFNM	**Special Fully Nested Mode:** 0 = Disabled by writing a 0 to this bit. 1 = Special fully nested mode is programmed.	0	WO
03	BUF	**Buffered Mode:** This bit must be programmed to 0 which is non-buffered mode. ***Note:*** Writing '1' will result in undefined behavior.	0	WO
02	MSBM	**Master/Slave in Buffered Mode:** Must be programmed to 0.	0	WO
01	AEOI	**Automatic End of Interrupt:** 0 = This bit must normally be programmed to 0. This is the normal end of interrupt. 1 = Automatic End of Interrupt (AEOI) mode is programmed. AEOI is discussed in Section 19.2.7.2.	0	WO
00	MM	**Microprocessor Mode:** This bit must be programmed to 1 to indicate that the controller is operating in an Intel Architecture-based system. ***Note:*** Writing '0' will result in undefined behavior.	1	WO

19.2.2.6 OCW1 – Operational Control Word 1 (Interrupt Mask)

Table 610. OCW1 – Operational Control Word 1 (Interrupt Mask)

I/O Address: Master Controller - 021h
Slave Controller - 0A1h

Size: 8 bits per controller

Default Value: *00h*

Bits	Name	Description	Reset Value	Access
07:00	IRM	**Interrupt Request Mask:** 0 = The corresponding IRQ mask bit is cleared and interrupt requests are again accepted by the controller. 1 = The corresponding IRQ line is masked. Masking IRQ2 on the master controller also masks the interrupt requests from the slave controller.	00h	RW

19.2.2.7 OCW2 – Operational Control Word 2

Following a device reset or ICW initialization, the controller enters the fully nested mode of operation. Non-specific EOI without rotation is the default. Both rotation mode and specific EOI mode are disabled following initialization.

Table 611. OCW2 – Operational Control Word 2

I/O Address: Master Controller - 020h Slave Controller - 0A0h			***Size:*** 8 bits per controller	
Default Value: Bits[07:05] = 001 Bits[04:00] = undefined				
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:05	REOI	**Rotate and EOI Codes (R, SL, EO):** These three bits control the Rotate and End of Interrupt modes and combinations of the two. 000 Rotate in Auto EOI Mode (Clear) 001 Non-specific EOI command 010 No Operation 011 *Specific EOI Command 100 Rotate in Auto EOI Mode (Set) 101 Rotate on Non-Specific EOI Command 110 *Set Priority Command 111 *Rotate on Specific EOI Command *L0 - L2 Are Used	001	WO
04:03	OCW2_SEL	**OCW2 Select:** When selecting OCW2, bits 04:03 = “00”	X	WO
02:00	INT_LS	**Interrupt Level Select:** L2, L1, and L0 determine the interrupt level acted upon when the SL bit is active. A simple binary code, outlined above, selects the channel for the command to act upon. When the SL bit is inactive, these bits do not have a defined function; programming L2, L1 and L0 to 0 is sufficient in this case. Code Interrupt Level CodeInterrupt Level 000I RQ0/8 100IRQ4/12 001 IRQ1/9 101 IRQ5/13 010 IRQ2/10 110IRQ6/14 011 IRQ3/11 111IRQ7/15	X	WO

19.2.2.8 OCW3 – Operational Control Word 3

Table 612. OCW3 – Operational Control Word 3 (Sheet 1 of 2)

I/O Address: Master Controller - 020h Slave Controller - 0A0h			***Size:*** 8 bits per controller	
Default Value: *001XX10b*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
07	Reserved	Reserved.	0	
06	SMM	**Special Mask Mode:** 0 = The Special Mask Mode will not be used by an interrupt service routine. 1 = The Special Mask Mode can be used by an interrupt service routine to dynamically alter the system priority structure while the routine is executing, through selective enabling/disabling of the other channel's mask bits. Bit 5, the ESMM bit, must be set for this bit to have any meaning.	0	WO
05	ESMM	**Enable Special Mask Mode:** 0 = Disable. The SMM bit becomes a “don't care”. 1 = Enable the SMM bit to set or reset the Special Mask Mode.	1	WO

Table 612. OCW3 – Operational Control Word 3 (Sheet 2 of 2)

I/O Address: Master Controller - 020h Slave Controller - 0A0h ***Default Value:*** *001XX10b*			***Size:*** 8 bits per controller	
Bits	**Name**	**Description**	**Reset Value**	**Access**
04:03	O3S	**OCW3 Select:** When selecting OCW3, bits04:03 = “01”.	X	WO
02	PMC	**Poll Mode Command:** 0 = Disable. Poll Command is not issued. 1 = Enable. The next I/O read to the interrupt controller is treated as an interrupt acknowledge cycle. An encoded byte is driven onto the data bus, representing the highest priority level requesting service.	X	WO
01:00	RRC	**Register Read Command:** These bits provide control for reading the In-Service Register (ISR) and the Interrupt Request Register (IRR). When bit 1=0, bit 0 does not affect the register read selection. When bit 1=1, bit 0 selects the register status returned following an OCW3 read. If bit 0=0, the IRR is read. If bit 0=1, the ISR is read. Following ICW initialization, the default OCW3 port address read is “read IRR”. To retain the current selection (read ISR or read IRR), always write a 0 to bit 1 when programming this register. The selected register can be read repeatedly without reprogramming OCW3. To select a new status register, OCW3 must be reprogrammed prior to attempting the read. 00 No Action 01 No Action 10 Read IRQ Register 11 Read IS Register	10	WO

19.2.2.9 Offset 4D0h: ELCR1 – Master Edge/Level Control

In edge mode, (bit[X] = 0), the interrupt is recognized by a low to high transition. In level mode (bit[X] = 1), the interrupt is recognized by a high level. The cascade channel, IRQ2, the heart beat timer (IRQ0), and the keyboard controller (IRQ1), cannot be put into level mode.

Table 613. Offset 4D0h: ELCR1 – Master Edge/Level Control (Sheet 1 of 2)

I/O Address: 4D0h ***Default Value:*** *00h*			***Size:*** 8 bit	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07	ECL7	**Edge Level Control IRQ7:** 0 = Edge mode. The interrupt is recognized by a low to high transition. 1 = Level mode. The interrupt is recognized by a high level.	0	RW
06	ECL6	**Edge Level Control IRQ6:** 0 = Edge mode. The interrupt is recognized by a low to high transition. 1 = Level mode. The interrupt is recognized by a high level.	0	RW
05	ECL5	**Edge Level Control IRQ5:** 0 = Edge mode. The interrupt is recognized by a low to high transition. 1 = Level mode. The interrupt is recognized by a high level.	0	RW

Table 613. Offset 4D0h: ELCR1 – Master Edge/Level Control (Sheet 2 of 2)

I/O Address: 4D0h ***Size:*** 8 bit

Default Value: *00h*

Bits	Name	Description	Reset Value	Access
04	ECL4	**Edge Level Control IRQ4:** 0 = Edge mode. The interrupt is recognized by a low to high transition. 1 = Level mode. The interrupt is recognized by a high level.	0	RW
03	ECL3	**Edge Level Control IRQ3:** 0 = Edge mode. The interrupt is recognized by a low to high transition. 1 = Level mode. The interrupt is recognized by a high level.	0	RW
02:00	Reserved	Reserved.	0	

19.2.2.10 Offset 4D1h: ELCR2 – Slave Edge/Level Control

In edge mode, (bit[X] = 0), the interrupt is recognized by a low to high transition. In level mode (bit[X] = 1), the interrupt is recognized by a high level. The real time clock, IRQ8#, and the floating point error interrupt, IRQ13, cannot be programmed for level mode.

Table 614. Offset 4D1h: ELCR2 – Slave Edge/Level Control

I/O Address: 4D1h ***Size:*** 8 bit

Default Value: *00*

Bits	Name	Description	Reset Value	Access
07	ECL15	**Edge Level Control IRQ15:** 0 = Edge 1 = Level	0	RW
06	ECL14	**Edge Level Control IRQ14:** 0 = Edge 1 = Level	0	RW
05	Reserved	Reserved.	0	
04	ECL12	**Edge Level Control IRQ12:** 0 = Edge 1 = Level	0	RW
03	ECL11	**Edge Level Control IRQ11:** 0 = Edge 1 = Level	0	RW
02	ECL10	**Edge Level Control IRQ10:** 0 = Edge 1 = Level	0	RW
01	ECL9	**Edge Level Control IRQ9:** 0 = Edge 1 = Level	0	RW
00	Reserved	Reserved. Must be zero.	0	

19.2.3 Interrupt Handling

19.2.3.1 Generating Interrupts

The PIC interrupt sequence involves three bits, from the IRR, ISR, and IMR, for each interrupt level. These bits are used to determine the interrupt vector returned and status of any other pending interrupts. These bits are defined in Table 615.

Table 615. Interrupt Handling

Bits	Name	Description
IRR	Interrupt Request Register	This bit is set on a low to high transition of the interrupt line in edge mode, and by an active high level in level mode. This bit is set whether or not the interrupt is masked. However, a masked interrupt will not generate INTR.
ISR	Interrupt Service Register	This bit is set, and the corresponding IRR bit cleared, when an interrupt acknowledge cycle is seen, and the vector returned is for that interrupt.
IMR	Interrupt Mask Register	Determines whether an interrupt is masked. Masked interrupts do not generate INTR.

19.2.3.2 Acknowledging Interrupts

The processor generates an interrupt acknowledge cycle that is translated into an Interrupt Acknowledge Special Cycle to the IICH. The PIC translates this cycle into two internal INTA# pulses expected by the 8259 cores. The PIC uses the first internal INTA# pulse to freeze the state of the interrupts for priority resolution. On the second INTA# pulse, the master or slave sends the interrupt vector to the processor with the acknowledged interrupt code. This code is based upon bits [07:03] of the corresponding ICW2 register, combined with three bits representing the interrupt within that controller.

Table 616. Content of Interrupt Vector Byte

Master, Slave Interrupt	Bits [07:03]	Bits [02:00]
IRQ7,15	ICW2[07:03]	111
IRQ6,14		110
IRQ5,13		101
IRQ4,12		100
IRQ3,11		011
IRQ2,10		010
IRQ1,9		001
IRQ0,8		000

19.2.3.3 Hardware/Software Interrupt Sequence

1. One or more of the Interrupt Request lines (IRQ) are raised high in edge mode, or seen high in level mode, setting the corresponding IRR bit.
2. The PIC sends INTR active (high) to the processor if an asserted interrupt is not masked.
3. The processor acknowledges the INTR and responds with an interrupt acknowledge cycle.

4. Upon observing the acknowledge cycle, it is converted into two cycles that the internal 8259 pair can respond to. Each cycle appears as an interrupt acknowledge pulse on the internal INTA# pin of the cascaded interrupt controllers.
5. Upon receiving the first internally generated INTA# pulse, the highest priority ISR bit is set and the corresponding IRR bit is reset. On the trailing edge of the first pulse, a slave identification code is broadcast internally by the master PIC to the slave PIC. The slave controller determines if it must respond with an interrupt vector during the second INTA# pulse.
6. Upon receiving the second internally generated INTA# pulse, the PIC returns the interrupt vector. If no interrupt request is present, the PIC will return vector 7 from the master controller.
7. This completes the interrupt cycle. In AEOI mode, the ISR bit is reset at the end of the second INTA# pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

19.2.4 Initialization Command Words (ICW)

Before operation can begin, each 8259 must be initialized. In the IICH, this is a four byte sequence. The four initialization command words are referred to by their acronyms: ICW1, ICW2, ICW3, and ICW4.

The base address for each 8259 initialization command word is a fixed location in the I/O memory space: 20h for the master controller, and A0h for the slave controller.

19.2.4.1 ICW1

An I/O write to the master or slave controller base address with data bit 4 equal to 1 is interpreted as a write to ICW1. Upon sensing this write, the IICH PIC expects three more byte writes to 21h for the master controller or A1h for the slave controller to complete the ICW sequence.

A write to ICW1 starts the initialization sequence during which the following automatically occur:

1. Following initialization, an interrupt request (IRQ) input must make a low-to-high transition to generate an interrupt.
2. The Interrupt Mask Register is cleared.
3. IRQ7 input is assigned priority 7.
4. The slave mode address is set to 7.
5. Special Mask Mode is cleared and Status Read is set to IRR.

19.2.4.2 ICW2

The second write in the sequence, ICW2, is programmed to provide bits [07:03] of the interrupt vector that are released during an interrupt acknowledge. A different base is selected for each interrupt controller.

19.2.4.3 ICW3

The third write in the sequence, ICW3, has a different meaning for each controller.

- For the master controller, ICW3 is used to indicate which IRQ input line is used to cascade the slave controller. Within the IICH, IRQ2 is used. Therefore, bit 2 of ICW3 on the master controller is set to a 1, and the other bits are set to 0's.
- For the slave controller, ICW3 is the slave identification code used during an interrupt acknowledge cycle. On interrupt acknowledge cycles, the master

controller broadcasts a code to the slave controller if the cascaded interrupt won arbitration on the master controller. The slave controller compares this identification code to the value stored in its ICW3. If it matches, the slave controller assumes responsibility for broadcasting the interrupt vector.

19.2.4.4 ICW4

The final write in the sequence, ICW4, must be programmed by both controllers. At the very least, bit 0 must be set to a 1 to indicate that the controllers are operating in an Intel Architecture-based system.

19.2.5 Operation Command Words (OCW)

These command words reprogram the Interrupt Controller to operate in various interrupt modes.

- OCW1 masks and unmasks interrupt lines.
- OCW2 controls the rotation of interrupt priorities when in rotating priority mode, and controls the EOI function.
- OCW3 sets up ISR/IRR reads, enables/disables the Special Mask Mode SMM, and enables/disables polled interrupt mode.

19.2.6 Modes of Operation

19.2.6.1 Fully Nested Mode

In this mode, interrupt requests are ordered in priority from 0 through 7, with 0 being the highest. When an interrupt is acknowledged, the highest priority request is determined and its vector is placed on the bus. Additionally, the ISR for the interrupt is set. This ISR bit remains set until:

- the processor issues an EOI command immediately before returning from the service routine; or
- if in AEOI mode, on the trailing edge of the second INTA# pulse.

While the ISR bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels generate another interrupt. Interrupt priorities can be changed in the rotating priority mode.

19.2.6.2 Special Fully Nested Mode

This mode is used in a system where cascading is used, and the priority has to be conserved within each slave. In this case, the special fully-nested mode is programmed to the master controller. This mode is similar to the fully-nested mode with the following exceptions:

- When an interrupt request from a certain slave is in service, this slave is not locked out from the master's priority logic and further interrupt requests from higher priority interrupts within the slave is recognized by the master and initiates interrupts to the processor. In normal nested mode, a slave is masked out when its request is in service.
- When exiting the Interrupt Service Routine, software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a Non-Specific EOI command to the slave and then reading its ISR. If it is 0, a non-specific EOI can also be sent to the master.

19.2.6.3 Automatic Rotation Mode (Equal Priority Devices)

In some applications, there are a number of interrupting devices of equal priority. Automatic rotation mode provides for a sequential eight-way rotation. In this mode, a device receives the lowest priority after being serviced. In the worst case, a device requesting an interrupt has to wait until each of seven other devices are serviced at most once.

There are two ways to accomplish automatic rotation using OCW2; the Rotation on Non-Specific EOI Command (R=1, SL=0, EOI=1) and the rotate in automatic EOI Mode which is set by (R=1, SL=0, EOI=0).

19.2.6.4 Specific Rotation Mode (Specific Priority)

Software can change interrupt priorities by programming the bottom priority. For example, if IRQ5 is programmed as the bottom priority device, then IRQ6 is the highest priority device. The Set Priority Command is issued in OCW2 to accomplish this, where: R=1, SL=1, and L0-L2 is the binary priority level code of the bottom priority device.

In this mode, internal status is updated by software control during OCW2. However, it is independent of the EOI command. Priority changes can be executed during an EOI command by using the Rotate on Specific EOI Command in OCW2 (R=1, SL=1, EOI=1 and LO-L2=IRQ level to receive bottom priority.

19.2.6.5 Poll Mode

Poll Mode can be used to conserve space in the interrupt vector table. Multiple interrupts that can be serviced by one Interrupt Service Routine do not need separate vectors if the service routine uses the poll command. Polled Mode can also be used to expand the number of interrupts. The polling interrupt service routine can call the appropriate service routine, instead of providing the interrupt vectors in the vector table. In this mode, the INTR output is not used and the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by software using a Poll Command.

The Poll command is issued by setting P=1 in OCW3. The PIC treats its next I/O read as an interrupt acknowledge, sets the appropriate ISR bit if there is a request, and reads the priority level. Interrupts are frozen from the OCW3 write to the I/O read. The byte returned during the I/O read will contain a ‘1’ in bit 7 if there is an interrupt, and the binary code of the highest priority level in bits 2:0.

19.2.6.6 Edge and Level Triggered Mode

In ISA (legacy) systems this mode is programmed using bit 3 in ICW1, which sets level or edge for the entire controller. In the IICH, this bit is disabled and a new register for edge and level triggered mode selection, per interrupt input, is included. This is the Edge/Level control Registers ELCR1 and ELCR2.

If an ELCR bit is ‘0’, an interrupt request is recognized by a low to high transition on the corresponding IRQ input. The IRQ input can remain high without generating another interrupt. If an ELCR bit is ‘1’, an interrupt request is recognized by a high level on the corresponding IRQ input and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued to prevent a second interrupt from occurring.

In both the edge and level triggered modes, the IRQ inputs must remain active until after the falling edge of the first internal INTA#. If the IRQ input goes inactive before this time, a default IRQ7 vector is returned.

19.2.7 End of Interrupt (EOI) operations

An EOI can occur in one of two fashions: by a command word write issued to the PIC before returning from a service routine, or automatically when AEOI bit in ICW4 is set to 1.

19.2.7.1 Normal EOI

In Normal EOI, software writes an EOI command before leaving the Interrupt Service Routine to mark the interrupt as completed. There are two forms of EOI commands: Specific and Non-Specific. When a Non-Specific EOI command is issued, the PIC clears the highest ISR bit of those that are set to 1. Non-Specific EOI is the normal mode of operation of the PIC within the IICH, as the interrupt being serviced currently is the interrupt entered with the interrupt acknowledge. When the PIC is operated in modes which preserve the fully nested structure, software can determine which ISR bit to clear by issuing a Specific EOI.

An ISR bit that is masked will not be cleared by a Non-Specific EOI if the PIC is in the Special Mask Mode. An EOI command must be issued for both the master and slave controller.

19.2.7.2 Automatic EOI Mode

In this mode, the PIC automatically performs a Non-Specific EOI operation at the trailing edge of the last interrupt acknowledge pulse. From a system standpoint, this mode must be used only when a nested multi-level interrupt structure is not required within a single PIC. The AEOI mode can only be used in the master controller.

19.2.8 Masking Interrupts

19.2.8.1 Masking on an Individual Interrupt Request

Each interrupt request can be masked individually by the Interrupt Mask Register (IMR). This register is programmed through OCW1. Each bit in the IMR masks one interrupt channel. Masking IRQ2 on the master controller masks all requests for service from the slave controller.

19.2.8.2 Special Mask Mode

Some applications may require an Interrupt Service Routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The Special Mask Mode enables all interrupts not masked by a bit set in the Mask Register. Normally, when an interrupt service routine acknowledges an interrupt without issuing an EOI to clear the ISR bit, the interrupt controller inhibits all lower priority requests. In the Special Mask Mode, any interrupts may be selectively enabled by loading the Mask Register with the appropriate pattern. The special Mask Mode is set by OCW3.SSMM and OCW3.SMM set, and cleared when OCW3.SSMM and OCW3.SMM are cleared.

19.2.9 Steering of PCI Interrupts

The IICH can be programmed to allow PIRQA#-PIRQH# to be internally routed to interrupts 3-7, 9-12, 14 or 15, through the PARC, PBRC, PCRC, PDRC, PERC, PFRC, PGRC, and PHRC registers, respectively, in Chapter 16.0, “Offset 60h: PARC – PIRQ[A,B,C,D,E,F,G,H] Routing Control Register”. The assignment is programmable

through the PIRQx Route Control registers, located at 60–63h and 68–6Bh in Device 31, Function 0. One or more PIRQx# lines can be routed to the same IRQx input. If interrupt steering is not required, the Route registers can be programmed to disable steering.

The PIRQx# lines are defined as active low, level sensitive to allow multiple interrupts on a platform to share a single line across the connector. When PIRQx# is routed to specified IRQ line, software must change the corresponding ELCR1 or ELCR2 register to level sensitive mode. The IICH internally inverts the PIRQx# line to send an active high level to the PIC. When a PCI interrupt is routed onto the PIC, the selected IRQ can no longer be used by an ISA (legacy) device. Internal sources of the PIRQs, including SCI and TCO interrupts, cause the external PIRQ to be asserted. The IICH receives the PIRQ input, like all of the other external sources, and routes it accordingly.

19.3 Advanced Interrupt Controller: APIC

In addition to the standard ISA (legacy)-compatible PIC described in the previous chapter, the IICH also incorporates the APIC.

19.3.1 Interrupt Handling

The I/O APIC handles interrupts very differently than the 8259. Briefly, these differences are:

- **Method of Interrupt Transmission.** The I/O APIC transmits interrupts through memory writes on the normal data path to the processor, and interrupts are handled without the need for the processor to run an interrupt acknowledge cycle.
- **Interrupt Priority.** The priority of interrupts in the I/O APIC is independent of the interrupt number. For example, interrupt 10 can be given a higher priority than interrupt 3.
- **More Interrupts.** The I/O APIC in the IICH supports a total of 24 interrupts.
- **Multiple Interrupt Controllers.** The I/O APIC architecture allows for multiple I/O APIC devices in the system with their own interrupt vectors.

19.3.2 PCI/PCI Express* Message-Based Interrupts

When external devices through PCI/PCI Express wish to generate an interrupt, they send the message defined in the *PCI Express Base Specification* for generating INTA# - INTD#. These are translated internal assertions/deassertions of INTA# - INTD#.

19.3.2.1 Front Side Bus Interrupt Delivery

The IICH requires that the I/O APIC deliver interrupt messages to the processor in a parallel manner, rather than using the I/O APIC serial scheme.

Delivery of interrupts is completed by the IICH writing to a memory location that is snooped by the processor. The processor snoops the cycle to know which interrupt goes active.

The following sequence is used:

1. When the IICH detects an interrupt event (active edge for edge-triggered mode or a change for level-triggered mode), it sets or resets the internal IRR bit associated with that interrupt.
2. Internally, the IICH requests to use the bus in a way that automatically flushes upstream buffers. This can be internally implemented similar to a DMA device request.

3. When granted the NSI, the IICH delivers the message by performing a write cycle to the appropriate address with the appropriate data. The address and data formats are described in Section 19.3.2.5.

19.3.2.2 Edge-Triggered Operation

In this case, the "Assert Message" is sent when there is an inactive-to-active edge on the interrupt.

19.3.2.3 Level-Triggered Operation

In this case, the "Assert Message" is sent when there is an inactive-to-active edge on the interrupt. If the interrupt is still active after the EOI, then another "Assert Message" is sent to indicate that the interrupt is still active.

19.3.2.4 Registers Associated with Front-Side Bus Interrupt Delivery

Capabilities Indication is the capability to support front-side bus interrupt delivery indicated via ACPI configuration techniques. This involves the BIOS creating a data structure that gets reported to the ACPI configuration software.

19.3.2.5 EOI

The mechanism by which the processor may generate an EOI is PCI Express EOI message.

The PCI Express EOI message is used by IA-32 processors. It is broadcast to the internal IOxAPIC and the downstream PCI-Express ports. The data of the EOI message is the vector. This value is compared with all the vectors inside the IOxAPIC, and any match causes RTE[X].RIRR to be cleared. See Section 14.1.6.9 for a description of the EOI vendor-specific message.

19.3.2.6 Interrupt Message Format

The Intel® 3100 Chipset writes the interrupt message internally as a 32-bit memory write cycle. It uses the following formats shown in Table 617 and Table 618 for the address and data.

Table 617. Interrupt Delivery Address Format

Bits	Description
31:20	Is always FEEh.
19:12	**Destination ID (DID):** This is the same as bits 63:56 of the I/O Redirection Table entry for the interrupt associated with this message.
11:04	**Extended Destination ID (EDID):** This is the same as bits 55:48 of the I/O Redirection Table entry for the interrupt associated with this message.
03	**Redirection Hint (DLM):** This bit is used by the processor host bridge to allow the interrupt message to be redirected. 0 = The message is delivered to the agent (processor) listed in bits 19:12. 1 = The message is delivered to an agent with a lower interrupt priority This can be derived from bits 10:8 in the Data Field (see below). The Redirection Hint bit is a 1 if bits 10:8 in the delivery mode field associated with corresponding interrupt are encoded as 001 (Lowest Priority). Otherwise, the Redirection Hint bit is 0
02	**Destination Mode (DSM):** This bit is used only when the Redirection Hint bit is set to 1. If the Redirection Hint bit and the Destination Mode bit are both set to 1, then the logical destination mode is used, and the redirection is limited only to those processors that are part of the logical group as based on the logical ID.
01:00	Will always be 00b.

Table 618. Interrupt Delivery Data Format

Bits	Description
31:16	Will always be 0000h.
15	**Trigger Mode:** 1 = Level,0 = Edge. Same as the corresponding bit in the I/O Redirection Table for that interrupt.
14	**Delivery Status:** 0 = Deassert 1 = Assert If using edge-triggered interrupts, then the bit will always be 1, since only the assertion is sent. If using level-triggered interrupts, then this bit indicates the state of the interrupt input.
13:12	Will always be 0h.
11	**Destination Mode:** 1= Logical. 0 = Physical. Same as the corresponding bit in the I/O redirection table
10:08	**Delivery Mode:** This is the same as the corresponding bits in the I/O Redirection Table for that interrupt. 000 = Fixed 100 = NMI 001 = Lowest Priority 101 = INIT 010 = SMI/PMI 110 = Reserved 011 = Reserved 111 = ExtINT
07:00	**Vector:** This is the same as the corresponding bits in the I/O Redirection Table for that interrupt.

19.3.3 Memory Registers

The APIC is accessed via an indirect addressing scheme. Two registers are visible by software for manipulation of most of the APIC registers. These registers are mapped into memory space. The registers are shown in Table 619.

Table 619. APIC Direct Registers (LPC I/F – D31, F0) Summary

Address	Symbol	Register Name/Function	Default	Access
FEC0_0000h	IDX	Index Register	00h	RW
FEC0_0010h	DAT	Data	00h	RW
FEC0_0040h	EOI	EOI Register	00h	WO

19.3.3.1 Address FEC00000h: IDX – Index Register

The Index register selects which APIC indirect register to be manipulated by software. The selector values for the indirect registers are listed in Table 622. Software programs this register to select the desired APIC internal register.

Table 620. Address FEC00000h: IDX – Index Register

I/O Address: FEC0_0000h ***Size:*** 8 bit

Default Value: *00h*

Bits	Name	Description	Reset Value	Access
07:00	APIC Index	**APIC Index:** This is an 8-bit pointer into the I/O APIC register table.	00h	RW

19.3.3.2 Address FEC00010h: DAT – Data Register

This register specifies the data to be read or written to the register pointed to by the Index register. This register can be accessed only in Dword quantities. The register is described in Section 19.3.3.4.

Table 621. Address FEC00010h: DAT – Data Register

I/O Address: FEC0_0010h ***Size:*** 32 bit

Default Value: *00h*

Bits	Name	Description	Reset Value	Access
31:00	WDW	This is a register for the data to be read or written to the APIC indirect register pointed to by the Index register (Table 623).	00h	RW

19.3.3.3 Address FEC00040h: EOI – EOI Register

When a write is issued to this register, the IOxAPIC checks the lower 8 bits written to this register, and compares it with the vector field for each entry in the I/O Redirection Table. When a match is found, RTE.RIRR for that entry is cleared. If multiple entries have the same vector, each of those entries has RTE.RIRR cleared. Only bits 07:00 are used and bits 31:08 are ignored.

Table 622. Address FEC00040h: EOI – EOI Register

I/O Address: FEC0_0040h ***Size:*** 32 bit

Default Value: *00h*

Bits	Name	Description	Reset Value	Access
31:08	Reserved	Reserved. Software must always write a value of 0 to these bits.	00h	
07:00	REC	**Redirection Entry Clear:** When a write is issued to this register, the I/O APIC checks this field, and compares it with the vector field for each entry in the I/O Redirection Table. When a match is found, the Remote_IRR bit for that I/O Redirection Entry is cleared.	00h	WO

19.3.3.4 Index Registers

Table 622 lists the registers which can be accessed within the APIC via the Index (IDX) register. When accessing these registers, accesses must be done a DWord at a time, otherwise unspecified behavior results. Software should not attempt to write to reserved registers. Some reserved registers may return non-zero values when read. For example, software must never access byte 2 from the Data Register before accessing bytes 0 and 1. The hardware does not attempt to recover from a bad programming model in this case.

Table 623. APIC Indirect Registers (LPC I/F – D31, F0) Summary

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00	00	ID	Identification Register	0000h	RW
01	01	VS	Version Register	00170020	RO
02	0F	—	Reserved		RO
10	11	REDIR_TBL0	Redirection Table 0 Register	XXXX0000_0001XXXX	RW, RO
12	13	REDIR_TBL1	Redirection Table 1 Register		RW, RO
…	…	—	—		—
3E	3F	REDIR_TBL23	Redirection Table 23 Register		RW, RO
40F	FF	—	Reserved		RO

19.3.3.5 Offset 00h: ID – Identification Register

The APIC ID serves as a physical name of the APIC. The APIC bus arbitration ID for the APIC is derived from its I/O APIC ID. This register is reset to 0 on power-up reset.

Table 624. Offset 00h: ID – Identification Register

I/O Address: 00h ***Size:*** 32 bit

Default Value: *0000h*

Bits	Name	Description	Reset Value	Access
31:28	Reserved	Reserved	0	
27:24	AID	**APIC Identification:** Software must program this value before using the APIC.	0	RW
23:16	Reserved	Reserved	0	
15	Scratchpad	Scratchpad	0	RW
14	Reserved	Reserved	0	
13:00	Reserved	Reserved	0	

19.3.3.6 Offset 01h: VS – Version Register

Each I/O APIC contains a hardwired Version Register that identifies different implementations of APIC and their versions. The maximum redirection entry information is also in this register. This lets the software know how many interrupt are supported by this APIC.

Table 625. Offset 01h: VS – Version Register

I/O Address: 01h ***Size:*** 32 bit

Default Value: *00170020h*

Bits	Name	Description	Reset Value	Access
31:24	Reserved	Reserved	0	
23:16	MRE	**Maximum Redirection Entries:** This is the entry number (0 being the lowest entry) of the highest entry in the redirection table. This field is hardwired to 17h to indicate 24 interrupts.	17h	RO
15	PRQ	**Pin Assertion Register Supported:** Indicate that the IOxAPIC does not implement the Pin Assertion Register.	0	RO
14:08	Reserved	Reserved	0	
07:00	VS	**Version:** Identifies the implementation version as IOxAPIC.	20h	RO

19.3.3.7 Offset 10-11h – 3E-3Fh: RTE[0-23] – Redirection Table Entry

The Redirection Table has a dedicated entry for each interrupt input pin. The information in the Redirection Table is used to translate the interrupt manifestation on the corresponding interrupt pin into an APIC message.

The APIC responds to an edge triggered interrupt as long as the interrupt is held until after the acknowledge cycle has begun. Once the interrupt is detected, a delivery status bit internal to the I/O APIC is set. The state machine steps ahead and waits for an acknowledgment from the APIC unit that the interrupt message was sent. Only then will the I/O APIC be able to recognize a new edge on that interrupt pin. That new edge only results in a new invocation of the handler if its acceptance by the destination APIC causes the Interrupt Request Register bit to go from 0 to 1. (In other words, if the interrupt was not already pending at the destination.) All bits are undefined except for bits 47:17 = 0 and bit 16 = 1.

Table 626. Offset 10-11h – 3E-3Fh: RTE[0-23] – Redirection Table Entry (Sheet 1 of 2)

I/O Address: 10-11h (vector 0) through 3E–3Fh (vector 23)

Size: 64 bit each, (accessed as two 32 bit quantities)

Default Value: *XXXX0000_0001XXXXh*

Bits	Name	Description	Reset Value	Access
63:56	DID	**Destination ID:** Destination ID of the local APIC.	X	RW
55:48	EDID	**Extended Destination ID:** Extended destination ID of the local APIC.	X	RW
47:17	Reserved	Reserved	0	
16	MSK	**Mask:** 0 = Not masked; an edge or level on this interrupt pin results in the delivery of the interrupt to the destination. 1 = Masked; interrupts are not delivered nor held pending. Setting this bit after the interrupt is accepted by a local APIC has no effect on that interrupt. This behavior is identical to the device withdrawing the interrupt before it is posted to the processor. It is software's responsibility to deal with the case where the mask bit is set after the interrupt message has been accepted by a local APIC unit but before the interrupt is dispensed to the processor.	1	RW
15	TM	**Trigger Mode:** This field indicates the type of signal on the interrupt pin that triggers an interrupt. 0 = Edge triggered 1 = Level triggered	X	RW
14	RIRR	**Remote IRR:** This bit is used for level triggered interrupts; its meaning is undefined for edge triggered interrupts. 0 = Reset when an EOI message matches the VCT field. 1 = Set when IOxAPIC sends the level interrupt message to the processor.	X	RW
13	POL	**Polarity:** This bit specifies the polarity of each interrupt input. 0 = Active high 1 = Active low	X	RW
12	DS	**Delivery Status:** This field contains the current status of the delivery of this interrupt. Writes to this bit have no effect. 0 = Idle. No activity for this interrupt. 1 = Pending. Interrupt has been injected but delivery is not complete. **For edge triggered interrupts**, this bit indicates that an event has occurred but an interrupt message has yet to be delivered to its targeted destination. Once the interrupt message is delivered, this bit will be cleared. **For level triggered interrupts**, this bit is set when the input event has occurred. This bit is cleared when the interrupt input event is removed. Note that as long as the interrupt input event is active, this bit remains active regardless of whether this interrupt has been delivered or not.	X	RO

Table 626. Offset 10-11h – 3E-3Fh: RTE[0-23] – Redirection Table Entry (Sheet 2 of 2)

I/O Address: 10-11h (vector 0) through 3E–3Fh (vector 23) ***Default Value:*** *XXXX0000_0001XXXXh*			***Size:*** 64 bit each, (accessed as two 32 bit quantities)	
Bits	**Name**	**Description**	**Reset Value**	**Access**
11	DSM	**Destination Mode:** This field determines the interpretation of the destination field. 0 = Physical. Destination APIC ID is identified by bits 59:56. 1 = Logical. Destinations are identified by matching bit 63:56 with the Logical Destination in the Destination Format Register and Logical Destination Register in each Local APIC.	X	RW
10:08	DLM	**Delivery Mode:** This field specifies how the APICs listed in the destination field must act upon reception of this signal. Delivery Modes will only operate as intended when used in conjunction with a specific trigger mode. The encodings are: 000 Fixed: Deliver the signal on the INTR signal of all processor cores listed in the destination. Trigger Mode can be edge or level. 001 Lowest Priority: Deliver the signal on the INTR signal of the processor core that is executing at the lowest priority among all the processors listed in the specified destination. Trigger Mode can be edge or level. 010 SMI: Not supported (see Section 19.3.4 for details). Requires the interrupt to be programmed as edge triggered. 011 Rreserved 100 NMI: Not supported (see Section 19.3.4 for details). 101 INIT: Not supported (see Section 19.3.4 for details). 110 Reserved 111 ExtINT: Deliver the signal to the INTR signal of all processor cores listed in the destination as an interrupt that originated in an externally connected 8259A compatible interrupt controller. The INTA cycle that corresponds to this ExtINT delivery is routed to the external controller that is expected to supply the vector. Requires the interrupt to be programmed as edge triggered.	X	RW
07:00	VCT	**Vector:** This field contains the interrupt vector for this interrupt. Values range between 10h and FEh.	X	RW

19.3.4 Supporting External Interrupt Controllers

The Intel® 3100 Chipset supports APICs off the PCI Express ports and does not support APICs on the PCI bus. EOI special cycles are forwarded to PCI Express ports but not the PCI bus.

19.4 PCI Interrupts via /PCI Express*

When external devices, attached through PCI Express wish to generate an interrupt, they send the message defined in the *PCI Express Base Specification* for generating INTA# - INTD#. These are translated into internal assertions/deassertions of INTA# - INTD#.

19.5 Serial Interrupt

19.5.1 Overview

The IICH interrupt controller supports a serial IRQ scheme. This allows a single signal to be used to report interrupt requests. The signal used to transmit this information is shared between the interrupt controller and all peripherals that support serial interrupts. The signal line, SERIRQ, is synchronous to the PCI clock, and follows the

sustained tri-state protocol that is used by legacy PCI signals. This means that if a device has driven SERIRQ low, it first drives it high synchronous to the PCI clock and releases it after the following PCI clock. The serial IRQ protocol defines this sustained tri-state signaling in the following fashion:

- **S - Sample Phase:** Signal driven low
- **R - Recovery Phase:** Signal driven high
- **T - Turn-around Phase:** Signal released

The IICH supports a message for 21 serial interrupts. These represent the 15 ISA (legacy) interrupts (IRQ0-1, 3-15), the four PCI interrupts, and the control signals SMI# and ISA (legacy) IOCHK#. The serial IRQ protocol does not support the additional APIC interrupts (20–23). The serial interrupt information is transferred using three types of frames:

- **Start Frame:** SERIRQ line driven low by the interrupt controller to indicate the start of IRQ transmission.
- **Data Frames:** IRQ information transmitted by peripherals. The interrupt controller supports 21 data frames.
- **Stop Frame:** SERIRQ line driven low by the interrupt controller to indicate end of transmission and next mode of operation.

19.5.2 Start Frame

The serial IRQ protocol has two modes of operation which affect the start frame:

- **Continuous Mode:** The interrupt controller is solely responsible for generating the start frame.
- **Quiet Mode:** Peripheral initiates the start frame, and the interrupt controller completes it.

The mode that must first be entered when enabling the serial IRQ protocol is continuous mode. In this mode, the IICH asserts the start frame. This start frame is four, six, or eight PCI clocks wide based upon the Serial IRQ Control Register(SCNT.SFPW) field, bits 01:00 at 64h in Device 31, Function 0 configuration space. This is a polling mode.

When the serial IRQ stream enters quiet mode (signaled in the Stop Frame), the SERIRQ line remains inactive and pulled up between the Stop and Start Frame until a peripheral drives the SERIRQ signal low. The IICH senses the line low and continues to drive it low for the remainder of the Start Frame. Since the first PCI clock of the start frame was driven by the peripheral in this mode, the IICH drives the SERIRQ line low for one PCI clock less than in continuous mode. This mode of operation allows for a quiet, and therefore lower power, operation.

19.5.3 Data Frames

Once the Start frame has been initiated, the SERIRQ peripherals start counting frames based on the rising edge of SERIRQ. Each of the IRQ/DATA frames has exactly three phases of one clock each. The three phases are:

- **Sample Phase:** During this phase, the SERIRQ device drives SERIRQ low if the corresponding interrupt signal is low. If the corresponding interrupt is high, then the SERIRQ devices tri-state the SERIRQ signal. The SERIRQ line remains high due to pull-up resistors (there is no internal pull-up resistor on this signal, an external pull-up resistor is required). A low level during the IRQ0–1 and IRQ2–15 frames indicates that an active-high ISA (legacy) interrupt is not being requested, but a low level during the PCI INT[A:D], SMI#, and ISA (legacy) IOCHK# frame indicates that an active-low interrupt is being requested.

- **Recovery Phase:** During this phase, the device drives the SERIRQ line high if in the Sample Phase it was driven low. If it was not driven in the sample phase, it is tri-stated in this phase.
- **Turn-around Phase:** The device tri-states SERIRQ.

19.5.4 Stop Frame

After all the data frames, a Stop Frame is driven by the IICH. The SERIRQ signal is driven low for two or three PCI clocks. The number of clocks is determined by the SERIRQ configuration register(SCNT.MD field in D31, F0 configuration space). The number of clocks determines the next mode as shown in Table 627.

Table 627. Stop Frame Definition

Stop Frame Width	Next Mode
2 PCI clocks	**Quite Mode:** Any SERIRQ device initiates a Start Frame.
3 PCI clocks	**Continuous Mode:** Only the IICH may initiate a Start Frame.

19.5.5 Serial Interrupts Not Supported via SERIRQ

There are three interrupts seen through the serial stream that are not supported by the IICH. These interrupts are generated internally, and are not sharable with other devices within the system. These interrupts are:

- IRQ0: .....Heartbeat interrupt generated off of the internal 8254 counter 0.
- IRQ8#: .....RTC interrupt can only be generated internally.
- IRQ13:.....Floating point error interrupt generated off of the processor assertion of FERR#.

The IICH ignores the state of these interrupts in the serial stream, and does not adjust their level based on the level seen in the serial stream.

19.5.6 Special Notes on IRQ14 and IRQ15

IRQ14 and IRQ15 are special interrupts, used by the SATA controller when it is not running in Native IDE mode. If in a legacy mode, IRQ14 and IRQ15 are not accepted from the serial stream, and instead come from these controllers. If the controllers are in Native mode, these interrupts are used by the interrupt controller.

19.5.7 Data Frame Format

Table 628 shows the format of the data frames for the associated interrupts. For the legacy (PCI) interrupts (A–D), the output from the IICH is ANDed with the legacy (PCI) input signal. This way, the interrupt can be signaled via both the legacy (PCI) interrupt input signal and via the SERIRQ signal (they are shared).

Table 628. Data Frame Format

Data Frame #	Interrupt	Clocks Past Start Frame	Comment
1	IRQ0	2	Ignored. IRQ0 can only be generated via the internal 8524.
2	IRQ1	5	Before port 60h latch.
3	SMI#	8	Causes SMI# if low. Sets bit 15 in the SMI_STS register.
4	IRQ3	11	

Table 628. Data Frame Format

5	IRQ4	14	
6	IRQ5	17	
7	IRQ6	20	
8	IRQ7	23	
9	IRQ8	26	Ignored. IRQ8# can only be generated internally or on ISA (legacy).
10	IRQ9	29	
11	IRQ10	32	
12	IRQ11	35	
13	IRQ12	38	Before port 60h latch
14	IRQ13	41	Ignored. IRQ13 can only be generated from FERR#.
15	IRQ14	44	Not attached to PATA or SATA logic.
16	IRQ15	47	Not attached to PATA or SATA logic.
17	IOCHCK#	50	Same as ISA (legacy) IOCHCK# going active.
18	PCI INTA#	53	Drive PIRQA#
19	PCI INTB#	56	Drive PIRQB#
20	PCI INTC#	59	Drive PIRQC#
21	PCI INTD#	62	Drive PIRQD#

20.0 Processor Interface

The IICH is expected to provide support for several different processors (see Table 4, “Supported Microprocessors” on page 50).

- The IICH interfaces to the processor with a variety of signals
- Standard Outputs to processor: A20M#, SMI#, NMI, INIT#, INTR, STPCLK#, IGNNE#, CPUSLP#

The IICH has many outputs to the processor that use standard (non open drain) buffers. The IICH has a separate Vcc signal which is pulled up at the system level to the processor voltage, and thus determines Voh for the outputs to the processor.

20.1 I/O Registers Associated with Processor Interface

Table 629. Processor I/F Signal State

Signal Name	S3 Hot	S3 Cold	S5
A20M#	Low	Off	Off
CPUSLP#	Low	Off	Off
IGNNE#	Low	Off	Off
INIT#	Low	Off	Off
INTR	Low	Off	Off
NMI	Low	Off	Off
SMI#	Low	Off	Off
STPCLK#	Low	Off	Off
FERR#	X	Low	Low

Notes:
X = Don't care
ND = Not Determined. May be high or low depending on programming.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 630. Processor I/F Registers Summary Table

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
61h	61h	NMI_SC	NMI Status and Control Register	00h	RW, RO
70h	70h	NMI_EN	NMI Enable (and Real Time Clock Index) Register	80h	RW (special)
92h	92h	PORT92	Fast A20 and Init Register	00h	RW
F0h	F0h	COPROC_ERR	Coprocessor Error Register	00h	WO
CF9	CF9h	RST_CNT	Reset Control Register	00h	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

20.1.1 Register Descriptions

20.1.1.1 Offset 61h: NMI_SC – NMI Status and Control Register

Table 631. Offset 61h: NMI_STS_CNT – NMI Status and Control Register

I/O Address: 61h	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07	SERR#_ NMI_STS	**SERR# NMI Source Status:** 0 = Bit is cleared when bit 2 is set to 1. 1 = Bit is set by any of the sources of the internal SERR on IICH; this includes SERR assertions forwarded from the secondary PCI bus, error from a PCI Express* port, Do_SERR or standard PCI Express error message from NSI, or internal Bus 0 functions that generate SERR#. Bit 2 must be cleared in this register in order for this bit to be set. This interrupt source is enabled by setting bit 2 to 0. To reset the interrupt, set bit 2 to 1 and then set it to 0. This bit is read-only. When writing to port 61h, this bit must be 0.	0b	RO
06	IOCHK_ NMI_STS	**IOCHK# NMI Source Status:** 0 = Bit is cleared when bit 3 is set to 1. 1 = Bit is set if a legacy agent (via SERIRQ) asserts ISA IOCHK# and bit 3 is cleared (IOCHK_NMI_EN). This interrupt source is enabled by setting bit 3 to 0. To reset the interrupt, set bit 3 to 1 and then set bit 3 to 0. When writing to port 61h, this bit must be a 0.	0b	RO
05	TMR2_ OUT_STS	**Timer Counter 2 OUT Status:** This bit reflects the current state of the 8254 counter 2 output. Counter 2 must be programmed following any PCI reset for this bit to have a determinate value. When writing to port 61h, this bit must be a 0.	0b	RO
04	REF_ TOGGLE	**Refresh Cycle Toggle:** This signal toggles from either 0 to 1 or 1 to 0 at a rate that is equivalent to when refresh cycles would occur. When writing to port 61h, this bit must be a 0. Assumed for compatibility, although no legacy refreshes occur. Must toggle at legacy refresh rate (every 15 µs).	0b	RO
03	IOCHK_ NMI_EN	**IOCHK# NMI Enable:** 0 = ISA IOCHK# NMIs are enabled. 1 = ISA IOCHK# NMIs are disabled and cleared.	0b	RW
02	PCI_SERR_E N	**PCI SERR# Enable:** 0 = SERR# NMIs are enabled. 1 = The SERR# NMIs are disabled and cleared.	0b	RW
01	SPKR_DAT_E N	**Speaker Data Enable:** 0 = The SPKR output is a 0. 1 = The SPKR output is equivalent to the Counter 2 OUT signal value.	0b	RW
00	TIM_CNT2_E N	**Timer Counter 2 Enable:** 0 = Counter 2 counting is disabled. 1 = Counting is enabled.	0b	RW

20.1.1.2 Offset 70h: NMI_EN – NMI Enable (and Real Time Clock Index) Register

This register is write-only for normal operation. In Alt-Access mode, this register can be read to find the NMI Enable status and the RTC index value.

Note: The RTC Index field is write-only for normal operation. This field can only be read in Alt-Access Mode. Note, however, that this register is aliased to Port 74h and all bits are readable at that address. See Chapter 21.0, Section 21.2, “RTC I/O Registers” on page 654 for more information.

Table 632. Offset 70h: NMI_EN – NMI Enable (and Real Time Clock Index) Register

I/O Address: 70h	***Size:*** 8 bit
Default Value: 80h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07	NMI_EN	**NMI Enable:** 0 = NMI sources are enabled. 1 = All NMI sources are disabled.	1b	RWS
06:00	RTC_INDX	**Real Time Clock Index (Address):**This data goes to the RTC to select which register or CMOS RAM address is being accessed.	0000000b	RWS

Note: Software must preserve the value of bit 7 at I/O addresses 70h. When writing to this address, software must first read the value, and then write the same value for bit 7 during the sequential address write. Note that port 70h is not directly readable. The only way to read this register is through Alt Access mode. Although RTC Index bits 06:00 are readable from port 74h, bit 7 will always return 0. If the NMI# enable is not changed during normal operation, software can alternatively read this bit once and then retain the value for all subsequent writes to port 70h.

20.1.1.3 Offset 92h: PORT92 – Fast A20 and Init Register

Table 633. Offset 92h: PORT92 – Fast A20 and Init Register

I/O Address: 92h	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:02	Reserved	Reserved.	00h	
01	ALT_A20_GATE	**Alternate A20 Gate:** 0 = The A20M# signal can potentially go active. This bit is ORed with the A20GATE input signal to generate A20M# to the processor. 1 = A20M# signal disabled.	0b	RW
00	INIT_NOW	**INIT# forced active:** 0 = INIT# is not forced to be active 1 = When this bit transitions from a 0 to a 1, it forces INIT# active for 16 PCI clocks.	0b	RW

20.1.1.4 Offset F0h: COPROC_ERR – Coprocessor Error Register

Table 634. Offset F0h: COPROC_ERR – Coprocessor Error Register

I/O Address: F0h	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	OPROC_ERR	Any value written to this register causes IGNNE# to go active, if FERR# generates an internal IRQ13. In order for FERR# to generate an internal IRQ13, the **Coprocessor Error Enable** bit (Section 14.1.6.9, "Offset 31FFh: OIC – Other Interrupt Control Register" on page 548, bit 1) must be set to 1.	00h	WO

20.1.1.5 Offset CF9h: RST_CNT – Reset Control Register

Table 635. Offset CF9h: RST_CNT – Reset Control Register

I/O Address: CF9h		***Size:*** 8 bit		
Default Value: 00h		***Power Well:*** Core		
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:04	Reserved	Reserved	00h	
03	FULL_RST	**Full Reset:** This bit is used to determine the states of SLP_S3#, SLP_S4# and SLP_S5# after a hard reset (not soft reset). 0 = SLP_S3#, SLP_S4# and SLP_S5# are kept high. 1 = Full reset, driving SLP_S3#, SLP_S4# and SLP_S5# low for 3–5 seconds if the following conditions are met: • SYS_RST = 1 (bit 1 of this register) Hard Reset not soft reset. • RST_CPU is written from 0 to 1(bit 2 of this register). • After PWROK going low (with RSMRST# high), or after two TCO timeouts. When this bit is set, it also causes the full power cycle (SLP_S3/4/5# assertion) in response to SYSRESET#, PWROK#, and Watchdog timer reset sources.	0b	RW
02	RST_CPU	**Reset Processor:** This bit causes either a hard or soft reset to the Processor depending on the state of the SYS_RST bit (bit 1 in this same register). Software causes the reset by setting this bit from a 0 to a 1.	0b	RW
01	SYS_RST	**System Reset:** This bit determines the type of reset caused via RST_CPU (bit 2 of this register). 0 = And RST_CPU goes from 0 to 1 (Soft Reset), then it forces INIT# active for 16 PCI clocks. 1 = And RST_CPU goes from 0 to 1 (Hard Reset), then it forces PLTRST# (and PCIRST#) and SUS_STAT# active for 5 to 6 ms. The IICH main power well is reset when this bit is 1 (and some suspend well logic may also be reset).	0b	RW
00	Reserved	Reserved	0b	

20.2 Processor Interface Signals

This section provides additional behavioral descriptions of the signals that interface between the IICH and the processor.

Note: The behavior of some signals may vary during processor reset, because the signals are used for frequency strapping.

Note: Some of the processor interface output signals are latched during STPCLK# assertion. See Chapter 22.0, “Power Management” for details.

20.2.1 A20M# (Mask A20)

The A20M# signal is active (low) when both of the following conditions are true:

1. The ALT_A20_GATE bit (Bit 1 of PORT92 register) is a ‘0’.

   and

2. The A20GATE input signal is a ‘0’.

The A20GATE input signal is expected to be generated by the external microcontroller (KBC).

20.2.2 INIT# (Initialization)

The INIT# signal is active (driven low) based on any one of several events described in Table 636. When any of these events occur, INIT# is driven low for 16 PCI clocks, then driven high.

Note: The 16-clock counter for INIT# assertion halts while STPCLK# is active. Thus, if INIT# is supposed to go active while STPCLK# is asserted, it goes active after STPCLK# goes inactive.

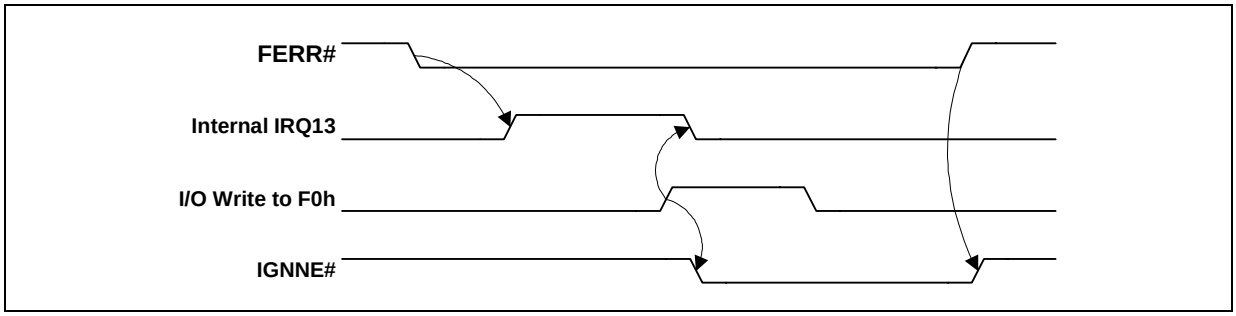
Table 636. INIT# Going Active

Cause of INIT# Going Active	Comment
Shutdown special cycle from processor observed on the IICH interconnect (from IMCH).	
PORT92 write, where INIT_NOW (bit 0) transitions from a 0 to a 1.	
PORTCF9 write, where SYS_RST (bit 1) was a 0 and RST_CPU (bit 2) transitions from 0 to 1.	
RCIN# input signal goes low. RCIN# is expected to be driven by the external microcontroller (KBC).	0 to 1 transition on RCIN# must occur before the IICH arms INIT# to be generated again.RCIN# signal is expected to be high during S3-hot and low (due to core power being off) during S3-cold and S5 states. Transitions on the RCIN# signal in those states (or in the transition to those states) may not necessarily cause the INIT# signal to be generated to the processor.
CPU BIST	In order to enter BIST, the software sets the CPU_BIST_EN bit, and then does a full processor reset using the CF9 register.

20.2.3 FERR# / IGNNE# (Coprocessor Error / Ignore Numeric Error)

The IICH supports the coprocessor error function with the FERR#/IGNNE# pins. The function is enabled via the COPROC_ERR_EN bit (Device 31, Function 0, Offset D0, Bit 13); refer to Table 571 for details. FERR# is tied directly to the Coprocessor Error signal of the processor. If FERR# is driven active by the processor, IRQ13 goes active (internally). When it detects a write to the COPROC_ERR register, the IICH negates the internal IRQ13 and drives IGNNE# active. IGNNE# remains active until FERR# is driven inactive. IGNNE# is never driven active unless FERR# is active.

Figure 81. Coprocessor Error Timing Diagram



If COPROC_ERR_EN is not set, then the assertion of FERR# does not generate an internal IRQ13, nor writes to F0h generate IGNNE#.

20.2.4 NMI (Non-Maskable Interrupt)

Non-Maskable Interrupts (NMIs) can be generated by several sources that are described in Table 637.

Table 637. NMI Sources

Cause of NMI	Comment
SERR# goes active (either internally, externally via SERR# signal, or via message from IMCH)	
ISA IOCHK# goes active via SERIRQ# stream (legacy system Error)	IOCHK# is a legacy signal. The Intel® 3100 Chipset does not have this pin, but it may be on the platform.
D30_SECSTS register (D30, F0, 1Eh), bit 8 (Detected parity error on PCI by North PCI unit)	Enabled by D30, F0, 04h, bit 6
D31F0_DEV_STS register (D31, F0, 06h), bit 8 (Detected parity error on PCI by South PCI unit)	Enabled by D30, F0, 04h, bit 6
Watch Dog Timer (LPC bus: logical device 6) first stage 35-bit Down Counter reaches zero.	Enabled by WDT_INT_TYPE field in the WDT Configuration Register.

20.2.5 INTR# (Interrupt Signals)

The behavior of the INTR signal and I/O APIC interrupt signals are described in Chapter 19.0, “Interrupts”.

20.2.6 STPCLK# and CPUSLP# (Stop Clock Request and Processor Sleep Signals)

These active-low signals are controlled by the power management logic. See Chapter 22.0, “Power Management” for more details.

20.2.7 Enhanced Intel SpeedStep Technology (EIST) Signals

Enhanced Intel SpeedStep Technology (EIST) is not supported.

20.2.8 DPSLP# (Deeper Sleep)

DPSLP# is not supported.

21.0 Real Time Clock (LPC I/F – D31: F0)

21.1 Overview

The Real Time Clock (RTC) module provides a battery backed-up date and time keeping device with two banks of static RAM with 128 bytes each. The first bank has 114 bytes for general purpose usage. Three interrupt features are available: time of day alarm with once a second to once a month range, periodic rates of 122 µs to 500 ms, and end of update cycle notification. Seconds, minutes, hours, days, day of week, month, and year are all counted. Daylight savings compensation is optional. The hour is represented in 12 or 24 hour format, and data can be represented in BCD or binary format. The design is meant to be functionally compatible with the Motorola MS146818B. The time keeping comes from a 32.768 KHz oscillating source, which is divided to achieve an update every second. The lower 14 bytes on the lower RAM block have very specific functions. The first ten are for time and date information. The next four (0Ah to 0Dh) are registers, which configure and report RTC functions.

The time and calendar data must match the data mode (BCD or binary) and hour mode (12 or 24 hour) as selected in register B Table 642. The programmer MUST make sure that data stored in these registers is within the reasonable values ranges and represents a possible date and time. The exception to these ranges is to store a value of C0 - FF in the alarm bytes to indicate a “don’t care” situation. All alarm conditions must match to trigger an Alarm Flag, which could trigger an Alarm Interrupt if enabled. The SET bit in register B must be ‘1’ while programming these locations to avoid clashes with update cycles. Access to time and date information is done through the RAM locations. If a RAM read from the ten time and date bytes is attempted during an update cycle, the value read will not necessarily represent the true contents of those locations. Any RAM writes under the same conditions are ignored.

Note: The leap year determination for adding a 29th day to February does not take into account the end-of-the-century exceptions. The logic simply assumes that all years divisible by four are leap years. According to the Royal Observatory Greenwich, years that are divisible by 100 are typically not leap years. In every fourth century (years divisible by 400, like 2000), the 100-year-exception is overridden and a leap-year occurs. Note that the year 2100 is the first time in which the current RTC implementation would incorrectly calculate the leap-year.

21.2 RTC I/O Registers

The RTC internal registers and RAM are organized as two banks of 128 bytes each, called the standard and extended banks. The first 14 bytes of the standard bank contain the RTC time and date information along with four registers, A–D, that are used for configuration of the RTC. The extended bank contains a full 128 bytes of battery backed SRAM, and is accessible even when the RTC module is disabled (via the RTC configuration register). Registers A – D do not physically exist in the RAM.

All data movement between the host CPU and the real-time clock is done through registers mapped to the standard I/O space. The register map appears below in Table 638.

Table 638. I/O Registers

I/O Locations	If U128E bit = 0	Function:
70h and 74h	Also alias to 72h and 76h	Real-Time Clock (Standard RAM) Index Register ***Note:*** Writes to 72h, 74h, and 76h do not affect NMI enable (bit 7 of 70h)
71h and 75h	Also alias to 73h and 77h	Real-Time Clock (Standard RAM) Target Register
72h and 76h		Extended RAM Index Register (if enabled)
73h and 77h		Extended RAM Target Register (if enabled)

I/O locations 70h and 71h are the standard legacy location for the real-time clock. The map for this bank is shown in Table 639. Locations 72h and 73h are for accessing the extended RAM. The extended RAM bank is also accessed using an indexed scheme. I/O address 72h is used as the address pointer and I/O address 73h is used as the data register. Index addresses above 127h are not valid. If the extended RAM is not needed, it may be disabled.

Note: Software must preserve the value of bit 7 at I/O addresses 70h. When writing to this address, software must first read the value, and then write the same value for bit 7 during the sequential address write. Note that port 70h is not directly readable. The only way to read this register is through Alt Access mode. Although RTC Index bits 06:00 are readable from port 74h, bit 7 will always return 0. If the NMI# enable is not changed during normal operation, software can alternatively read this bit once and then retain the value for all subsequent writes to port 70h.

21.3 Indexed Registers

The RTC contains two sets of indexed registers that are accessed using the two separate Index and Target registers (70/71h or 72/73h), as shown in Table 639.

Table 639. RTC (Standard) RAM Bank

Index	Name
00h	Seconds
01h	Seconds Alarm
02h	Minutes
03h	Minutes Alarm
04h	Hours
05h	Hours Alarm
06h	Day of Week
07h	Day of Month
08h	Month
09h	Year
0Ah	Register A
0Bh	Register B
0Ch	Register C
0Dh	Register D
0Eh – 7Fh	114 Bytes of User RAM

21.3.1 Register Descriptions

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

21.3.1.1 RTC_REGA – Register A (General Configuration)

Table 640. Indexed Registers Summary

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
0Ah	0Ah	RTC_REGA	This register is used for general configuration of the RTC functions.	XXX	RW
0Bh	0Bh	RTC_REGB	This register is used for general configuration of the RTC functions.	X0X00XXX	RW
0Ch	0Ch	RTC_REGC	This register is used for general configuration of the RTC functions.	00X00000b (X: Undefined)	RO
0Dh	0Dh	RTC_REGD	This register is used for general configuration of the RTC functions.	10XXXXXXb (X: Undefined)	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Table 641. RTC_REGA – Register A (General Configuration) (Sheet 1 of 2)

I/O Address: 0Ah ***Size:*** 8 bit

Default Value: XXX ***Power Well:*** RTC

Bits	Name	Description	Reset Value	Access
07	UIP	**Update in progress:** This bit may be monitored as a status flag. 0 = The update cycle will not start for at least 488 µs. The time, calendar, and alarm information in RAM is always available when the UIP bit is 0. 1 = The update is soon to occur or is in progress.	X	RW
06:04	DV[2:0]	**Division Chain Select:** These three bits control the divider chain for the oscillator, and are not affected by RSMRST# or any other reset signal. DV[2] corresponds to bit 6. **DV2 DV1 DV0 Function** 0 1 0 Normal Operation 1 1 X Divider Reset 0 0 1 Invalid 0 0 0 Invalid	X	RW

Table 641. RTC_REGA – Register A (General Configuration) (Sheet 2 of 2)

I/O Address: 0Ah		***Size:*** 8 bit		
Default Value: XXX		***Power Well:*** RTC		
Bits	**Name**	**Description**	**Reset Value**	**Access**
03:00	RS[3:0]	**Rate Select:** Selects one of 13 taps of the 15 stage divider chain. The selected tap can generate a periodic interrupt if the PIE bit is set in Register B. Otherwise this tap sets the PF flag of Register C. If the periodic interrupt is not to be used, these bits should all be set to zero. RS3 corresponds to bit 3. **RS3 RS2 RS1 RS0Periodic Rate** 0 0 0 0Interrupt never toggles 0 0 0 13.90625 ms 0 0 1 07.8125 ms 0 0 1 1122.070 ms 0 1 0 0244.141 ms 0 1 0 1488.281 ms 0 1 1 0976.5625ms 0 1 1 11.953125 ms 1 0 0 03.90625 ms 1 0 0 17.8125 ms 1 0 1 015.625 ms 1 0 1 131.25 ms 1 1 0 062.5 ms 1 1 0 1125 ms 1 1 1 0250 ms 1 1 1 1500 ms	X	RW

21.3.1.2 RTC_REGB – Register B (General Configuration)

Table 642. RTC_REGB – Register B (General Configuration)

I/O Address: 0Bh ***Size:*** 8 bit

Default Value: X0X00XXX ***Power Well:*** RTC

Bits	Name	Description	Reset Value	Access
07	SET	**Update Cycle Inhibit:** Enables/Inhibits the update cycles. 0 = Update cycle occurs normally once each second. 1 = A current update cycle aborts and subsequent update cycles do not occur until SET is returned to zero. The BIOS may initialize time and calendar bytes safely. ***Note:*** This bit is not affected by RSMRST# nor any other reset signal. ***Note:*** Software must ensure this bit transitions from '1' to '0' once whenever the RTC coin battery is inserted. This is to ensure that the internal RTC time updates occur properly.	X	RW
06	PIE	**Periodic Interrupt Enable:** 0 = Disabled 1 = Allows an interrupt to occur with a time base set with the RS bits of register A. ***Note:*** This bit is cleared by RSMRST#, but not on any other reset.	0	RW
05	AIE	**Alarm Interrupt Enable:** 0 = Disabled 1 = the Alarm Interrupt Enable (AIE) bit allows an interrupt to occur when the AF is one as set from an alarm match from the update cycle. An alarm can occur once a second, one an hour, once a day, or once a month. ***Note:*** This bit is cleared by RTEST#, but not on any other reset.	X	RW
04	UIE	**Update-ended Interrupt Enable:** 0 = Disabled 1 = Allows an interrupt to occur when the update cycle ends. ***Note:*** This bit is cleared by RSMRST#, but not on any other reset.	0	RW
03	SQWE	**Square Wave Enable:** The Square Wave Enable bit serves no function in this device, but it is in this register bank to provide compatibility with the Motorola 146818B. There is not a SQW pin on this device. ***Note:*** This bit is cleared by RSMRST#, but not on any other reset.	0	RW
02	DM	**Data Mode:** The Data Mode (DM) bit specifies either binary or BCD data representation. 0 = denotes BCD 1 = denotes binary This bit is not affected by RSMRST# nor any other reset signal.	X	RW
01	HOURFORM	**Hour Format:** This bit indicates the hour byte format. 0 = Twelve-hour mode is selected. In twelve hour mode, the seventh bit represents AM as zero and PM as one. 1 = Twenty-four hour mode is selected. This bit is not affected by RSMRST# nor any other reset signal.	X	RW
00	DSE	**Daylight Savings Enable:** 0 = Disabled 1 = Triggers two special hour updates per year when set to one. One is on the first Sunday in April, where time increments from 1:59:59 AM to 3:00:00 AM. The other is the last Sunday in October when the time first reaches 1:59:59 AM, it is changed to 1:00:00 AM. The time must increment normally for at least two update cycles (seconds) previous to these conditions for the time change to occur properly. These special update conditions do not occur when the DSE bit is set to zero. The days for the hour adjustment are those specified in United States federal law as of 1987, which is different than previous years. ***Note:*** This bit is not affected by RSMRST# nor any other reset signal.	X	RW

21.3.1.3 RTC_REGC – Register C (Flag Register)

Table 643. RTC_REGC – Register C (Flag Register)

I/O Address: 0Ch ***Size:*** 8 bit
Default Value: 00X00000b (X: Undefined) ***Power Well:*** RTC

Bits	Name	Description	Reset Value	Access
07	IRQF	**Interrupt Request Flag:** Interrupt Request Flag = (PF * PIE) + (AF * AIE) + (UF *UFE). This also causes the RTC Interrupt to be asserted. ***Note:*** This bit is cleared upon RSMRST# or a read of Register C.	0b	RO
06	PF	**Periodic Interrupt Flag:** 0 = No taps are specified. 1 = Set when the tap as specified by the RS bits of register A is one. ***Note:*** This bit is cleared upon RSMRST# or a read of Register C.	0b	RO
05	AF	**Alarm Flag:** 0 = All Alarm values do not match the current time 1 = All Alarm values match the current time. ***Note:*** This bit is cleared upon RTEST# or a read of Register C.	X	RO
04	UF	**Update-ended Flag:** 0 = Update cycle not detected 1 = This bit is set immediately following an update cycle for each second. ***Note:*** The bit is cleared upon RSMRST# or a read of Register C.	0b	RO
03:00	Reserved	**Reserved**	000b	

21.3.1.4 RTC_REGD – Register D (Flag Register)

Table 644. RTC_REGD – Register D (Flag Register)

Offset: 0Dh ***Size:*** 8 bit
Default Value: 10XXXXXXb (X: Undefined) ***Power Well:*** RTC

Bits	Name	Description	Reset Value	Access
07	VRT	**Valid RAM and Time Bit:** This bit is hard-wired to '1' in the RTC power well. This bit should always be written as a 0 for write cycle, however it will return a 1 for read cycles.	1b	RW
06	Reserved	**Reserved:** This bit always returns a 0 and should be set to 0 for write cycles.	0b	
05:00	DA	**Date Alarm:** These bits store the date of month alarm value. If set to 000000, then a don't care state is assumed. The host must configure the dates alarm for these bits to do anything, yet they can be written at any time. If the date alarm is not enabled, these bits return zeros to mimic the functionality of the Motorola 146818B. These bits are not affected by any reset assertion.	XXXXXXb	RW

21.4 Update Cycles

An update cycle occurs once a second, if the SET bit of register B is not asserted and the divide chain is properly configured. During this procedure, the stored time and date are incremented, overflow is checked, a matching alarm condition is checked, and the time and date are rewritten to the RAM locations. The update cycle will start at least 488 µs after the UIP bit of register A is asserted, and the entire cycle will not take more than 1984 µs to complete. The time and date RAM locations (0–9) is disconnected from the external bus during this time.

To avoid update and data corruption conditions, external RAM access to these locations can safely occur upon the detection of either of two conditions.

1. When an updated-ended interrupt is detected, almost 999 ms is available to read and write the valid time and date data.
2. If the UIP bit of Register A is detected to be low, there is at least 488 µs before the update cycle begins.

Warning: The overflow conditions for leap years and daylight savings adjustments are based on more than one date or time item. To ensure proper operation when adjusting the time, the new time and data values should be set at least two seconds before one of these conditions (leap year, daylight savings time adjustments) occurs.

21.5 Interrupts

The real-time clock interrupt is internally routed within the IICH both to the I/O APIC and the 8259. It is mapped to interrupt vector 8. This interrupt does not leave the IICH prior to connection to the interrupt controller, nor is it shared with any other interrupt. IRQ8# from the SERIRQ stream is ignored. However, the High Performance Event Timers can also be mapped to IRQ8#; in this case, the RTC interrupt is blocked.

21.6 Lockable RAM Ranges

The RTC's battery-backed RAM supports two 8-byte ranges that can be locked via the PCI configuration space. If the locking bit is set, the corresponding range in the RAM is not readable or writeable. A write cycle to those locations has no effect. A read cycle to those locations does not return the locations actual value (resultant value is undefined).

Once a range is locked, the range can be unlocked only by a hard reset, which will invoke the BIOS and allow it to relock the RAM range.

21.7 Century Rollover

The IICH detects a roll over when the Year byte (RTC I/O space, index offset 09h) transitions from 99 to 00 (e.g., a rollover from December 31, 2099, 11:59:59 p.m. to 12:00:00 a.m on January 1st, 2100). Upon detecting the rollover, the IICH sets the NEWCENTURY_STS bit (TCOBASE + 04h, bit 7). If the system is in an S0 state, this causes an SMI#. The SMI# handler can update registers in the RTC RAM that are associated with the century value.

If the system is in a sleep state (S3 and S5) when the century rollover occurs, the IICH also sets the NEWCENTURY_STS bit, but no SMI# is generated. When the system resumes from the sleep state, BIOS should check the NEWCENTURY_STS bit and update the century value in the RTC RAM.

21.8 Month and Year Alarms

This function is not supported.

22.0 Power Management

22.1 The Features

Note: The following list is for informational purposes, and must not be used by designers or validators as part of the behavioral description.

- ACPI Power and Thermal Management Support
  - Processor THRMTRIP# emergency shutdown
  - ACPI 24-Bit Timer
  - Software initiated throttling of processor performance for Thermal and Power Reduction
  - SCI and SMI# Generation
- PCI PME# Signal for Wake Up from Low-Power states
- PCI Express WAKE# signal for Wake from Low-Power states
- SYS_Reset# input to eliminate external glue logic
- System Clock Control
  - ACPI C0 state . Full On: Processor operating. Individual devices may be shut off to save power.
  - ACPI C1 state . Auto-Halt: Processor has executed a AutoHalt instruction and is not executing code. The processor snoops the bus and maintains cache coherency.
  - ACPI C2 state .....Stop-Grant state (using STPCLK# signal) halts processor's instruction stream.
- System Sleeping State Control
  - ACPI S0 state – All power planes active (awake)
  - ACPI S3 state – Suspend to RAM (STR)
  - Supports both S3-Cold and S3-Hot state. There are no behavioral differences between the two states from Intel® 3100 Chipset.
  - ACPI S5 state – Soft Off (SOFF)
  - Power Failure Detection and Recovery
- Streamlined Legacy Power Management Support for APM-Based Systems
- Support for several different processors (see Table 4, "Supported Microprocessors" on page 50)

22.2 System Power States

Table 645 shows the power states defined for Intel® 3100 Chipset-based platforms, as well as the target power consumption by the System and Intel® 3100 Chipset itself. The state names generally match the corresponding ACPI states.

Table 645. General Power States and Consumption for Systems

State / Substates	Legacy Name / Description
G0/S0/C0	**Full On:** Processor operating. Individual devices may be shut to save power. The different processor operating levels are defined by Cx states, as shown in Table 646. Within the C0 state, Intel® 3100 Chipset can throttle the processor using the STPCLK# signal to reduce processor power consumption. The throttling can be initiated by software or by the operating system or BIOS.
G0/S0/C1	**Auto-Halt:** Processor has executed a AutoHalt instruction and is not executing code. The processor snoops the bus and maintains cache coherency.
G0/S0/C2	**Stop-Grant:** The STPCLK# signal goes active to the processor. The processor performs a Stop-Grant cycle, halts its instruction stream, and remains in that state until the STPCLK# signal goes inactive. In the Stop-Grant state, the processor snoops the bus and maintains cache coherency. ***Note:*** This state is not supported for IA-64 processors. They must instead use C1.
G1/S3	**Suspend-To-RAM (STR):** The system context is maintained in system DRAM, but power is shut to non-critical circuits. Memory is retained, and refreshes continue. All clocks stop except RTC. ***Note:***
G2/S5	**Soft Off (SOFF):** System context is not maintained. All power is shut except for the logic required to restart. A full boot is required when waking.
G3	**Mechanical OFF (MOFF):** System context not maintained. All power is shut off except for the RTC. No "Wake" events are possible, because the system does not have any power. This state occurs if the user removes the batteries, turns off a mechanical switch, or if the system power supply is at a level that is insufficient to power the "waking" logic. When system power returns, transition will depend on the state just prior to the entry to G3.

Table 646 shows the transition rules among the various states.

Note: Transitions among the various states may appear to temporarily transition through intermediate states. These intermediate transitions and states are not listed in Table 646.

Table 646. State Transition Rules (Sheet 1 of 2)

Present State	Transition Trigger	Next State *
G0/S0/C0	Processor halt instruction	G0/S0/C1
	Level 2 Read	G0/S0/C2
	SLP_EN bit set	G1/S3 or G2/S5 state (specified by SLP_TYP)
	Power Button Override	G2/S5
	Mechanical Off/Power Failure	G3
G0/S0/C1	Any Enabled Break Event	G0/S0/C0
	STPCLK# goes active	G0/S0/C2
	Power Button Override	G2/S5
	Resume Power Failure	G3
G0/S0/C2	Any Enabled Break Event	G0/S0/C0
	Power Button Override	G2/S5
	Resume Power Failure	G3
G1/S3,	Any Enabled Wake Event	G0/S0/C0
	Power Button Override	G2/S5
	Resume Power Failure	G3

Table 646. State Transition Rules (Sheet 2 of 2)

Present State	Transition Trigger	Next State *
G2/S5	Any Enabled Wake Event	G0/S0/C0
	Resume Power Failure	G3
G3	Power Returns	Option to go to S0/C0 (reboot) or G2/S5 (stay off until power button pressed or other enabled wake event). Some wake events are preserved through a power failure.

22.3 Power Planes

22.3.1 System Power Planes

The system has several independent power planes, as described in Table 647.

Note: When a particular power plane is shutdown, it must go to a 0 V level.

Table 647. System Power Planes

Plane	Controlled By	Description
Processor	SLP_S3# signal	The SLP_S3# signal can be used to cut the processor's power completely.
MAIN	SLP_S3#or SLP_S4# signal	S3-Cold: When SLP_S3# goes active, power can be shutdown to any circuit not required to wake the system from the S3 state. Since the S3 state requires that the memory context be preserved, power must be retained to the main memory. Devices on the PCI bus, LPC Interface, NSI and PCI Express will typically be shut off when the Main power plane is shut off, although they may have small subsections powered. S3-Hot: The S3-Hot state keeps more of the platform logic, including the Intel® 3100 Chipset core well, powered to reduce the cost of external power plane logic.SLP_S4# is used to cut the main power well, rather than using SLP_S3#. SLP_S3# is only used to shut system clocks. This impacts the board design, but there is no specific Intel® 3100 Chipset bit or strap needed to indicate which option is selected.
DEVICES and MEMORY	SLP_S5# signal	When SLP_S5# goes active, power can be shut off to any circuit not required to wake the system from the S5 state. Since the memory context does not need to be preserved in the S5 state, the power to the memory can also be shut off.
DEVICE[*n*]	GPIO	Individual subsystems may have their own power plane. For example, GPIO signals may be used to control the power to disk drives, audio amplifiers, or the display screen

22.3.2 Power Planes

Although power planes are not specific, there are many interface signals that go to devices that may be powered down. They include:

— USB .....Intel® 3100 Chipset can tri-state USB output signals and shut off input buffers if USB wakeup is not desired

— PCI Express.....See Section 30.3 for details on the PCI Express ports and when they are powered off/on.

Note: Intel® 3100 Chipset's signal 3.3SUS must be valid at least five microseconds before the power to 1.5SUS reaches 0.7 V.

22.4 IMCH-IICH Messages

Intel® 3100 Chipset supports the NSI messaging protocol. Messages associated with power management and state transitions are summarized in Table 648. The Legacy Protocol messages are shown only for reference. NSI

Table 648. IMCH-IICH Messages

Legacy Message	New Message	Direction	Description/Comment:
---	Reset-Warn	IICH→IMCH	Warning from the IICH to the IMCH that the IICH is about to assert the PLTRST# signal. The IMCH is expected to acknowledge this with the Reset-Warn-Ack. However, it the IMCH fails to do this within the timeout period, the IICH will assert the reset.
---	Reset-Warn-Ack	IMCH→IICH	Acknowledge from the IMCH that it has seen the Reset-Warn message and is now ready for the IICH to cause the reset.
Stop-Grant	Stop-Grant (REQ-C2)	IMCH→IICH	If the processor is in C0 - indication that the processor has issued last Stop-Grant cycle. The IMCH may receive more than one Stop-Grant cycle from the processor(s). It is the IMCH's responsibility to only send the last Stop-Grant down to Intel® 3100 Chipset.
Go-C0	Go-C0	IICH→IMCH	Indication that the system is going back to C0 state.
--	Ack-C0	IMCH→IICH	Acknowledge that IMCH observed the Go-C0 message and is ready to proceed.
--	Go-C2	IICH→IMCH	This is an indication that the processor has been put into Stop-Grant state. When coming from C0, this tells the IMCH that it is safe to assert SLP#.
--	Ack-C2	IMCH→IICH	IMCH indicates it observed the Go-C2 message and is now ready to proceed.If going toward C0, the IICH is free to deassert STPCLK#.
Go-C3	Go-S3	IICH→IMCH	Indication that the IICH is getting ready to put the system into S3 or S5 state.
Ack-C3	Ack-S3	IMCH→IICH	Indication that the IMCH observed the Go-S3 message and is ready to proceed.
--	REQ-C0 (Break-Ind)	IMCH →IICH	This is an indication from the IMCH to the IICH that the IMCH thinks the processor must be brought to a C0 state. This would be sent for several cases: 1. If the IMCH had received a "Pending Break Event" indication from the processor. This is needed when PBE# is not muxed with FERR# and is instead muxed with some pin that goes only to the IMCH. 2. The IMCH has some internal device, or link to external device, that can cause a break event that is not associated with an interrupt. ***Note:*** The IMCH is not required to implement this message.

22.5 Power Management Registers

This section shows Intel® 3100 Chipset's power management registers. The power management registers are distributed within the PCI Device 31: Function 0 space, as well as a separate I/O range. Each register is described below. Unless otherwise indicated, bits are in the main (core) power well.

Bits not explicitly defined in each register are assumed to be reserved. Software should not attempt to use the value read from a reserved bit, because it may not be consistently 1 or 0.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 649. Summary Table for Power Management PCI Registers (PM — D31:F0)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
A0h	A1h	GEN_PMCON_1	General Power Management Configuration 1 Register (Core Well)	0200h	RW, RO, RWO
A2h	A2h	GEN_PMCON_2	General Power Management Configuration 2 Register (Resume Well)	00h	RW, RWC
A4h	A4h	GEN_PMCON_3	General Power Management Configuration 3 Register (RTC Well)	00h	RW, RWC
B8h	BBh	GPI_ROUT	GPI Route Control Register	00000000h	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

22.5.1 Power Management Registers in PCI Device 31, Function 0

22.5.1.1 Offset A0h: GEN_PMCON_1 – General PM Configuration 1 Register

Table 650. Offset A0h: GEN_PMCON_1 – General PM Configuration 1 Register (Sheet 1 of 2)

Device: 31
Function: 0
Offset: *A0h - A1h*
Size: *16-bit*
Default Value: *0200h*
Power Well: *Bits 10,07:00 - Core, Bit 9 - Resume*
Lockable: *No*
Usage: *ACPI, Legacy*

Bits	Name	Description	Reset Value	Access
15:11	**Reserved**	**Reserved.**	000h	
10	BIOS_PCI_EXP_EN	This bit acts as a global enable for the SCI associated with the PCI Express* ports. 0 = The various PCI Express ports and cannot cause the PCI_EXP_STS bit to go active. 1 = The various PCI Express ports and can cause the PCI_EXP_STS bit to go active.	0	RW
09	PWRBTN_LVL	This bit indicates the current state of the PWRBTN# signal. 0 = Low 1 = High	1	RO
08:07	**Reserved**	**Reserved.**	00	
06	64_EN	Software sets this bit to indicate that the processor is an IA-64 processor, not an IA-32 processor. This may be used in various state machines where there are behavioral differences. 0 = IA-32 processor 1 = IA-64 processor	0	RW
05	Reserved	Reserved.	0	

Table 650. Offset A0h: GEN_PMCON_1 – General PM Configuration 1 Register (Sheet 2 of 2)

Device: 31 **Function:** 0
Offset: A0h - A1h **Size:** 16-bit
Default Value: 0200h **Power Well:** Bits 10,07:00 - Core, Bit 9 - Resume
Lockable: No
Usage: ACPI, Legacy

Bits	Name	Description	Reset Value	Access
04	SMI_LOCK	When this bit is set, writes to the GLB_SMI_EN bit will have no effect. Once the SMI_LOCK bit is set, writes of 0 to SMI_LOCK bit will have no effect (i.e., once set, this bit can only be cleared by PLTRST#).	0	RWO
03:02	Reserved	Reserved	00	
01:00	PER_SMI_SEL	Software sets these bits to control the rate at which the periodic SMI# is generated: 00 = 64 seconds (default) 01 = 32 seconds 10 = 16 seconds 11 = 8 seconds Tolerance for the timer is ±1 second.	00	RW

22.5.1.2 Offset A2h: GEN_PMCON_2 – General PM Configuration 2 Register

Table 651. Offset A2h: GEN_PMCON_2 – General PM Configuration 2 Register (Sheet 1 of 2)

Device: 31 **Function:** 0
Offset: A2h **Size:** 8-bit
Default Value: 00h **Power Well:** Resume
Lockable: No
Usage: ACPI, Legacy

Bits	Name	Description	Reset Value	Access
07	DIB	DRAM Initialization bit: This bit does not effect hardware functionality in any way. BIOS is expected to set this bit prior to starting the DRAM initialization sequence and to clear this bit after completing the DRAM initialization sequence. BIOS can detect that a DRAM initialization sequence was interrupted by a reset by reading this bit during the boot sequence. If the bit is 1, then the DRAM initialization was interrupted. See Section 22.7.1 for the expected BIOS response. This bit is reset by the assertion of the RSMRST# pin.	0	RW
06:05	Reserved	Reserved	00	
04	SRS	**System Reset Status:** 0 = SYS_RESET# button Not pressed. 1 = Intel® 3100 Chipset sets this bit when the SYS_RESET# button is pressed. BIOS is expected to read this bit and clear it if it is set. ***Note:*** This bit is also reset by RSMRST# and CF9h resets.	0	RWC

Table 651. Offset A2h: GEN_PMCON_2 – General PM Configuration 2 Register (Sheet 2 of 2)

Device: 31	**Function:** 0
Offset: A2h	**Size:** 8-bit
Default Value: 00h	**Power Well:** Resume
Lockable: No	
Usage: ACPI, Legacy	

Bits	Name	Description	Reset Value	Access
03	CTS	Processor Thermal Trip Status: 0 = Software clears this bit by writing a 1 to it. 1 = This bit is set when PLTRST# is inactive and THRMTRIP# goes active while the system is in an S0 state. ***Note:*** This bit is also reset by RSMRST# and CF9h resets. It is not reset by the shutdown and reboot associated with the CPUTHRMTRIP# event.	0	RWC
02	MAWVS	Minimum SLP_S4# Assertion Width Violation Status: 0 = Software clears this bit by writing a 1 to it. 1 = Hardware sets this bit when the SLP_S4# assertion width is less than the time programmed in the SLP_S4# Minimum Assertion Width field (D31.F0.A4h.5:4). When exiting G3, Intel® 3100 Chipset begins the timer when the RSMRST# input deasserts. ***Note:*** This bit is functional regardless of the value in the SLP_S4# Assertion Stretch Enable. This bit is reset by the assertion of the RSMRST# pin, but can be set in some cases before the default value is readable.	0	RWC
01	CPUPWR_FLR	Processor Power Failure: 0 = Software (typically) BIOS clears this bit by writing a 0 to it. 1 = Indicates that the VRMPWRGD input signal from the processor's VRM went low. ***Note:*** VRMPWRGD is sampled using the RTC clock. Therefore, low times that are less than one RTC clock period may not be detected by Intel® 3100 Chipset.	0	RW
00	PWROK_FLR	Power OK Failure: 0 = Software clears this bit by writing a 1 to it, or when the system goes into a G3 state. 1 = This bit will be set any time PWROK goes low, when the system was in S0 state. The bit will be cleared only by software by writing a 1 to this bit or when the system goes to a G3 state. See Section 22.8.3 for more details about the PWROK pin functionality. ***Note:*** In the case of true PWROK failure, PWROK will go low first before the VRMPWRGD.	0	RWC

22.5.1.3 Offset A4h: GEN_PMCON_3 – General PM Configuration 3 Register

Table 652. Offset A4h: GEN_PMCON_3 – General PM Configuration 3 Register (Sheet 1 of 2)

Device: 31	**Function:** 0
Offset: A4h	**Size:** 8-bit
Default Value: 00h	**Power Well:** RTC
Lockable: No	
Usage: ACPI, Legacy	

Bits	Name	Description	Reset Value	Access
07:06	SWSMI_ RATE_SEL	This 2-bit value indicates when the SWSMI timer will time out. Valid values are: • 00 1.5 ms +/- 0.6 ms • 0116 ms +/- 4 ms • 10 32 ms +/- 4 ms • 11 64 ms +/- 4 ms These bits are not cleared by any type of reset except RTEST#.	00	RW
05:04	SMAW	**SLP_S4# Minimum Assertion Width:** This 2-bit value indicates the minimum assertion width of the SLP_S4# signal to guarantee that the DRAMs have been safely power-cycled. This value may be modified per platform depending on DRAM types, power supply capacitance, etc. Valid values are: • 111 to 2 seconds • 10 2 to 3 seconds • 01 3 to 4 seconds • 00 4 to 5 seconds This value is used in two ways: 1. If the SLP_S4# assertion width is ever shorter than this time, a status bit (D31.F0.A2h.2) is set for BIOS to read when S0 is entered 2. If enabled by bit 3 in this register, the hardware will prevent the SLP_S4# signal from deasserting within this minimum time period after asserting ***Note:*** The logic that measures this time is in the suspend power well. Therefore, when leaving the G3 state, the minimum time is measured from the deassertion of RSMRST#. RTEST# forces this field to the conservative default state (00b).	00	RW
03	SASE	SLP_S4# Assertion Stretch Enable: 0 = The SLP_S4# minimum assertion time is 1 to 2 RTCCLK. 1 = The SLP_S4# signal will minimally assert for the time specified in bits 5:4 of this register. This bit is cleared by RTEST#.	0	RW
02	RPS	RTC Power Status: 0 = RTEST# OK 1 = RTEST# indicates a weak or missing battery. The bit remains set until the software clears it by writing a 0 back to this bit position. This bit is not cleared by any type of reset.	X	RW

Table 652. Offset A4h: GEN_PMCON_3 – General PM Configuration 3 Register (Sheet 2 of 2)

Device: 31	**Function:** 0
Offset: A4h	**Size:** 8-bit
Default Value: 00h	**Power Well:** RTC
Lockable: No	
Usage: ACPI, Legacy	

Bits	Name	Description	Reset Value	Access
01	PWR_FLR	PWR_FLR: 0 = Indicates that the trickle current has not failed since the last time the bit was cleared. 1 = Indicates that the trickle current (from the main battery or trickle supply) was removed or failed. Software writes a 1 to this bit to clear it. This bit is in the RTC well, and is not cleared by any type of reset except RTEST#. ***Notes:*** 1. RSMRST# is sampled using the RTC clock. Therefore, low times that are less than one RTC clock period may not be detected by Intel® 3100 Chipset. 2. Clearing CMOS in Intel® 3100 Chipset platforms can be done by using a jumper on RTEST# or GPI, or using SAFEMODE strap. Implementations must not attempt to clear CMOS by using a jumper to pull VccRTC low.	0	RWC
00	AG3E	**AFTERG3_EN:** Determines what state to go to when power is reapplied after a power failure (G3 state). 0 = System will return to an S0 state (boot) after power is reapplied. 1 = System will return to the S5 state. In addition to software writes, this bit is set by the following hardware conditions: • Power Button Override • SMBus Unconditional Powerdown Message • Catastrophic Temperature condition from an internal sensor • Assertion of Processor Thermal Trip input	0	RW

22.5.1.4 Offset B8h: GPI_ROUT – GPI Routing Control Register

Table 653. Offset B8h: GPI_ROUT – GPI Routing Control Register

Device: 31	***Function:*** 0
Offset: *B8h – BBh*	***Size:*** *32-bit*
Default Value: *00000000h*	***Power Well:*** *Resume*
Lockable: *No*	
Usage: *ACPI or Legacy*	

Bits	Name	Description	Reset Value	Access
31:02	GPI	GPI[15] through GPI[1]: See bits 1:0 for description.	0000000h	RW
01:00	GPI[0] Route	If the corresponding GPIO is implemented and is set to an input, a '1' in the GP_LVL bit can be routed to cause an SMI# or SCI. If the GPIO is not set to an input, this field has no effect. • 00 – No effect (or GPIO unimplemented) • 01 – SMI# (if corresponding ALT_GPI_SMI_EN bit also set) • 10 – SCI (if corresponding GPE0_EN bit also set) • 11 – Reserved If the system is in an S3, S5 state and if the GPE0_EN bit is also set, then the GPI can cause a Wake event, even if the GPI is NOT routed to cause an SMI# or SCI. Exception: If the system is in S5 state due to a powerbutton override, then the GPIs will not cause wake events. ***Note:*** Core well GPIs are not capable of waking the system from sleep states where the core well is not powered.	00	RW

22.5.2 APM I/O Decode

Table 654 shows the I/O registers associated with APM support. This register space is enabled in the PCI Device 31: Function 0 space (APMC_EN), and cannot be moved (fixed I/O location).

Table 654. APM Register Map

Address	Mnemonic	Register Name	Default	Type
B2h	APM_CNT	Advanced Power Management Control Port	00h	RW
B3h	APM_STS	Advanced Power Management Status Port	00h	RW

22.5.2.1 Offset B2h: APM_CNT – Advanced Power Management Control Port Register

Used to pass an APM command between the OS and the SMI handler. Writes to this port not only store data in the APMC register, but also generates an SMI# when the APMC_EN bit is set.

Table 655. Offset B2h: APM_CNT – Advanced Power Management Control Port Register

I/O Address: B2h	**Size:** 8-bit
Default Value: 00h	**Power Well:** Core
Lockable: No	
Usage: Legacy Only	

Bits	Name	Description	Reset Value	Access
07:00	APM_CNT	Used to pass an APM command between the OS and the SMI handler. Writes to this port not only store data in the APMC register, but also generates an SMI# when the APMC_EN bit is set.	00h	RW

22.5.2.2 Offset B3h: APM_STS – Advanced Power Management Status Port Register

Used to pass data between the OS and the SMI handler. Basically, this is a scratchpad register and is not effected by any other register or function (other than a PCI reset).

Table 656. Offset B3h: APM_STS – Advanced Power Management Status Port Register

I/O Address: B3h	**Size:** 8-bit
Default Value: 00h	**Power Well:** Core
Lockable: No	
Usage: Legacy Only	

Bits	Name	Description	Reset Value	Access
07:00	APM_STS	Used to pass data between the OS and the SMI handler. Basically, this is a scratchpad register and is not affected by any other register or function (other than a platform reset).	00h	RW

22.5.3 General I/O Decode Ranges for Power Management

Table 657 shows the registers associated with ACPI and Legacy power management support. These registers are enabled in the PCI Device 31: Function 0 space (ACPI Enable in Section 16.2.2.2), and can be moved to any I/O location (128-byte aligned) determined by ABASE in Section 16.2.2.1 (referenced in this chapter by PMBASE). The registers are defined to be compliant with the *Advanced Configuration and Power Interface (ACPI) Specification, Rev. 2.0*, and generally use the same bit names. All reserved bits and registers will always return 0 when read, and will have no effect when written.

Table 657. ACPI and Legacy I/O Register Map (Sheet 1 of 2)

PMBASE+ Offset	Register Name	ACPI Pointer	Default	Attributes
00 - 01h	PM1 Status	PM1a_EVT_BLK	0000h	Read/Write-Clear
02 - 03h	PM1 Enable	PM1a_EVT_BLK+2	0000h	Read/Write
04 - 07h	PM1 Control	PM1a_CNT_BLK	00000000h	Read/Write, Write-Only
08 - 0Bh	PM1 Timer	PMTMR_BLK	00000000h	Read-Only
0C - 0Fh	Reserved			
10h - 13h	Processor Control	P_BLK	0000h	Read/Write

Table 657. ACPI and Legacy I/O Register Map (Sheet 2 of 2)

PMBASE+ Offset	Register Name ACPI Pointer	Default	Attributes
14h	Level 2 Register P_BLK+4	0	Read-Only
15h	Reserved		
16h	Reserved		
17 - 1Fh	Reserved		
20h	Reserved		
28 - 2Bh	General Purpose Event 0 Status GPE0_BLK	00000000h	Read/Write-Clear
2C - 2Fh	General Purpose Event 0 Enables GPE0_BLK+4	00000000h	Read/Write
30 - 33h	SMI# Control and Enable	00000000h	Read/Write, Write-Only
34 - 37h	SMI Status Register	00000000h	Read/Write-Clear, Read-Only
38 - 39h	Alternate GPI SMI Enable Register	0000h	Read/Write
3A - 3Bh	Alternate GPI SMI Status Register	0000h	Read/Write-Clear
3Ch - 3Fh	Reserved		
42h	Reserved		
44 - 45h 4Ch - 4Dh	Device Trap Status Reserved	0000h	Read/Write
4Eh - 4Fh	Reserved		
50h	Reserved		
51h - 53h	Reserved		
54h - 57h	Reserved.		
58h - 5Fh	Reserved		
60 - 7Fh	Reserved for TCO Registers. See Chapter 27.0.		

Warning: Table 657 is for informational purposes, and must not be used by designers or validators as part of the behavioral description.

22.5.3.1 Offset 00h: PM1_STS – Power Management 1 Status Register

Table 658. Offset 00h: PM1_STS – Power Management 1 Status Register (Sheet 1 of 2)

I/O Address: PMBASE + 00h (ACPI PM1a_EVT_BLK)	**Size:** 16-bit
Default Value: 0000h	**Power Well:** Bits 0-7: Core, Bits 8-15: Resume (except 11 in RTC)
Lockable: No	
Usage: ACPI or Legacy	

Bits	Name	Description	Reset Value	Access
15	WAK_STS	0 = Software clears this bit by writing a 1 to it. 1 = This bit can only be set by hardware when the system is in one of the Sleep states (via the SLP_EN bit) and an enabled Intel® 3100 Chipset Wake event occurs. Upon setting this bit, Intel® 3100 Chipset will transition the system to the ON state. This bit is not affected by hard resets caused by a CF9 write, but is reset by RSMRST#. If a power failure occurs (such as removed batteries) without the SLP_EN bit set, the WAK_STS bit will not be set when the power returns if the AFTER_G3 bit is 0. If the AFTER_G3 bit is 1, then the WAK_STS bit will be set after waking from a power failure. If necessary, the BIOS can clear the WAK_STS bit in this case.	0	RWC
14	Reserved	Reserved.	0	
13:12	Reserved	Reserved	00	
11	PRBTNOR_STS	0 = Software clears this bit by writing a 1 to it. 1 = This bit is set any time a Power Button Override Event occurs (i.e., the power button is pressed for at least 4 consecutive seconds), or due to the corresponding bit in the SMBus slave message, or due to an internal thermal sensor catastrophic condition. These events cause an unconditional transition to the S5 state, as well as sets the AFTERG3 bit. The BIOS or SCI handler clears this bit by writing a 1 to it. This bit is not affected by hard resets via CF9h writes, and is not reset by RSMRST#. Thus, this bit is preserved through power failures.	0	RWC
10	RTC_STS	0 = Software clears this bit by writing a 1 to it. 1 = Set when the RTC generates an alarm (assertion of the IRQ8# signal), and is not affected by any other enable bit. See RTC_EN for the effect when RTC_STS goes active. This bit is only set by hardware and can only be reset by writing a one to this bit position. This bit is not affected by hard resets caused by a CF9 write, but is reset by RSMRST#. Additionally if the RTC_EN bit (PMBASE + 02h, bit 10) is set, the setting of the RTC_STS bit will generate a wake event.	0	RWC
09	Reserved	Reserved	0	
08	PWRBTN_STS	This bit is set when the PWRBTN# signal is asserted (low), independent of any other enable bit. See PWRBTN_EN for the effect when PWRBTN_STS goes active. PWRBTN_STS is always a wake event. This bit is only set by hardware and can be cleared by software writing a one to this bit position. This bit is not affected by hard resets caused by a CF9 write, but is reset by RSMRST#. If the PWRBTN# signal is held low for more than 4 seconds, Intel® 3100 Chipset clears the PWRBTN_STS bit, sets the PWRBTNOR_STS bit, the system transitions to the S5 state, and only PWRBTN# is enabled as a wake event. If PWRBTN_STS bit is cleared by software while the PWRBTN# pin is still held low, this will not cause the PWRBTN_STS bit to be set. The PWRBTN# signal must go inactive and active again to set the PWRBTN_STS bit. ***Note:*** The SMBus Unconditional Power down message, the Processor Thermal Trip and the Internal Thermal Sensors' Catastrophic Condition result in behavior matching the Powerbutton Override, which includes clearing this bit.	0	RWC
07:06	Reserved	Reserved	0	

Table 658. Offset 00h: PM1_STS – Power Management 1 Status Register (Sheet 2 of 2)

I/O Address: *PMBASE + 00h (ACPI PM1a_EVT_BLK)* **Size:** *16-bit*

Default Value: *0000h* **Power Well:** *Bits 0-7: Core, Bits 8-15: Resume (except 11 in RTC)*

Lockable: *No*

Usage: *ACPI or Legacy*

Bits	Name	Description	Reset Value	Access
05	GBL_STS	0 = The SCI handler must then clear this bit by writing a 1 to the bit location. 1 = Set when an SCI is generated due to BIOS wanting the attention of the SCI handler. BIOS has a corresponding bit, BIOS_RLS, which will cause an SCI and set this bit. This bit will not cause wake events or SMI#. This bit is not effected by SCI_EN. ***Note:*** GBL_STS being set will cause an SCI, even if the SCI_EN bit is not set. Software must take great care not to set the BIOS_RLS bit (which causes GBL_STS to be set) if the SCI handler is not in place.	0	RWC
04:01	Reserved	Reserved	000	
00	TMROF_ STS	0 = The SCI or SMI# handler clears this bit by writing a 1 to the bit location. 1 = This bit gets set any time bit 22 of the 24-bit timer goes low (bits are numbered from 0 to 23). This will occur every 2.3435 seconds. When the TMROF_EN bit (PMBASE + 02h, bit 0) is set, then the setting of the TMROF_STS bit will additionally generate an SCI or SMI# (depending on the SCI_EN).	0	RWC

22.5.3.2 Offset 02h: PM1_EN – Power Management 1 Enables Register

Table 659. Offset 02h: PM1_EN – Power Management 1 Enables Register (Sheet 1 of 2)

I/O Address: *PMBASE + 02h (ACPI PM1a_EVT_BLK + 2)* **Size:** *16-bit*

Default Value: *0000h* **Power Well:** *Bits 0-7: Core, Bits 8-15: Resume*

Lockable: *No*

Usage: *ACPI or Legacy*

Bits	Name	Description	Reset Value	Access
15	Reserved	Reserved	0	
14	Reserved	Reserved.	0	
13:11	Reserved	Reserved	000	
10	RTC_EN	This is the RTC alarm enable bit. It works in conjunction with the SCI_EN bit: RTC_EN SCI_EN Effect when RTC_STS is set 0 X No SMI# or SCI. If system was in S3 or S5, no wake event occurs. 1 0 SMI#. If system was in S3 or S5, then a wake event occurs before the SMI#. 1 1 SCI. If system was in S3 or S5, then a wake event occurs before the SMI#.	0	RW
09	Reserved	Reserved	0	

Table 659. Offset 02h: PM1_EN – Power Management 1 Enables Register (Sheet 2 of 2)

I/O Address: PMBASE + 02h (ACPI PM1a_EVT_BLK + 2)
Size: 16-bit
Default Value: 0000h
Power Well: Bits 0-7: Core, Bits 8-15: Resume
Lockable: No
Usage: ACPI or Legacy

Bits	Name	Description	Reset Value	Access
08	PWRBTN_EN	This bit is the power button enable. It works in conjunction with the SCI_EN bit: PWRBTN_EN SCI_EN Effect when PWRBTN_STS is set 0 X No SMI# or SCI 1 0 SMI# 1 1 SCI ***Note:*** PWRBTN_EN has no effect on the PWRBTN_STS bit being set by the assertion of the power button. The Power Button is always enabled as a Wake event.	0	RW
07:06	Reserved	Reserved	00	
05	GBL_EN	The global enable bit. When both the GBL_EN and the GBL_STS are set, Intel® 3100 Chipset generates an SCI. 0 = Disable. 1 = Enable SCI on GBL_STS going active.	0	RW
04:01	Reserved	Reserved	00	
00	TMROF_EN	This is the timer overflow interrupt enable bit. It works in conjunction with the SCI_EN bit: TMROF_EN SCI_EN Effect when TMROF_STS is set 0 X No SMI# or SCI. If system was in S3 or S5, no wake event. 1 0 SMI#. If system was in S3 or S5, then a wake event occurs before the SMI#. 1 1 SCI. If system was in S3 or S5, then a wake event occurs before the SMI#.	0	RW

22.5.3.3 Offset 04h: PM1_CNT – Power Management 1 Control Register

Table 660. Offset 04h: PM1_CNT – Power Management 1 Control Register

I/O Address: PMBASE + 04h (ACPI PM1a_CNT_BLK)
Size: 32-bit
Default Value: 0000h
Power Well: Bits 0-7: Core, Bits 8-12: RTC, Bits 13-15: Resume
Lockable: No
Usage: ACPI or Legacy

Bits	Name	Description	Reset Value	Access
31:14	Reserved	Reserved	0	
13	SLP_EN	This is a write-only bit and reads to it always return a zero. Setting this bit causes the system to sequence into the Sleep state defined by the SLP_TYP field.	0	WO
12:10	SLP_TYP	This 3-bit field defines the type of Sleep the system must enter when the SLP_EN bit is set to 1. These bits are reset by RTEST# only. Bits — *Mode* — Typical Mapping *000* — *ON* — *S0* *001* — Reserved *010* — Reserved 011 — Reserved 100 — Reserved 101 — Suspend-To-RAM — S3 110 — Reserved 111 — Soft Off — S5	0	RW
09:03	Reserved	Reserved	0	
02	GBL_RLS	This bit is used by the ACPI software to raise an event to the BIOS software. BIOS software has a corresponding enable and status bits to control its ability to receive ACPI events. This bit always reads as 0.	0	WO
	Reserved	Reserved		
00	SCI_EN	Selects the SCI interrupt or the SMI# for various events. 0 = These events generate an SMI#. 1 = The events generate an SCI interrupt.	0	RW

22.5.3.4 Offset 08h: PM1_TMR – Power Management 1 Timer Register

Table 661. Offset 08h: PM1_TMR – Power Management 1 Timer Register

I/O Address: PMBASE + 08h (ACPI PMTMR_BLK) ***Size:*** 32 –bit
Default Value: 00000000h ***Power Well:*** Core
Lockable: No
Usage: ACPI

Bits	Name	Description	Reset Value	Access
31:24	Reserved	Reserved. Will always read as 0.	00h	
23:00	TMR_VAL	**Timer Value:** This read-only field returns the running count of the PM timer. This counter runs off a 3.579545 MHz clock (derived from 14.31818 MHz divided by 4). It is reset (to 0) during a PCI reset, and then continues counting as long as the system is in the S0 state. Anytime bit 22 of the timer goes HIGH to LOW (bits referenced from 0 to 23), the TMROF_STS bit is set. The High-to-Low transition will occur every 2.3435 seconds. Writes to this register have no effect.	00h	RO

22.5.3.5 Offset 10h: PROC_CNT – Processor Control Register

Table 662. Offset 10h: PROC_CNT – Processor Control Register (Sheet 1 of 3)

I/O Address: PMBASE + 10h (ACPI P_BLK) ***Size:*** 32-bit
Default Value: 00000000h ***Power Well:*** Core
Lockable: No (Except bits 7:5 are write-once).
Usage: ACPI or Legacy

Bits	Name	Description	Reset Value	Access
31:18	Reserved	Reserved.	00h	
17	THTL_STS	**Throttle STS:** 0 = No clock throttling is occurring (maximum processor performance). 1 = Indicates that the clock state machine is throttling the processor performance. This could be due to the THT_EN bit or the FORCE_THTL bit being set.	0	RO
16:09	Reserved	Reserved.	00h	
08	FORCE_THTL	**Force Throttle:** Software can set this bit to 1 to force the throttling. 0 = The throttling (at a duty cycle specified in THRM_DTY) does not start immediately and does generate an SMI#. 1 = The throttling (at a duty cycle specified in THRM_DTY) starts immediately and does not generate an SMI#.	0	RW

Table 662. Offset 10h: PROC_CNT – Processor Control Register (Sheet 2 of 3)

I/O Address: PMBASE + 10h (ACPI P_BLK) **Size:** 32-bit
Default Value: 00000000h **Power Well:** Core
Lockable: No (Except bits 7:5 are write-once).
Usage: ACPI or Legacy

Bits	Name	Description	Reset Value	Access
07:05	THRM_DTY	**Thermal Duty:** This write-once 3-bit field determines the duty cycle of the throttling when the FORCE_THTL bit is set. The duty cycle indicates the approximate percentage of time the STPCLK# signal is asserted while in the thermal throttle mode. The STPCLK# throttle period is 1024 PCICLKs. Throttling only occurs if the system is in the C0 state. If in the C2 state, no throttling occurs. Once the THRM_DTY field is written, subsequent writes have no effect until PLTRST# goes active. THRM_DTY Bits[2:0] / Throttle Mode / PCI Clocks *000* / Default (will be 50%) / 512 *001* / 87.5% / 896 *010* / 75.0% / 768 011 / 62.5% / 640 100 / 50% / 512 101 / 37.5% / 384 110 / 25% / 256 111 / 12.5% / 128	000	RW

Table 662. Offset 10h: PROC_CNT – Processor Control Register (Sheet 3 of 3)

I/O Address: PMBASE + 10h (ACPI P_BLK) ***Size:*** 32-bit
Default Value: 00000000h ***Power Well:*** Core
Lockable: No (Except bits 7:5 are write-once).
Usage: ACPI or Legacy

Bits	Name	Description	Reset Value	Access
04	THT_EN:	**Throttle Enable:** When this bit is set and the system is in a C0 state, it enables a software controlled STPCLK# throttling. The duty cycle is selected in the THTL_DTY field. 0 = Disable 1 = Enable	0	RW
03:01	THTL_DTY	**Throttle Duty:** This 3 –bit field determines the duty cycle of the throttling when the THT_EN bit is set. The duty cycle indicates the approximate percentage of time the STPCLK# signal is **asserted (low)** while in the throttle mode. The STPCLK# throttle period is 1024 PCICLKs. THRM_DTY Bits[3:0] / Throttle Mode / PCI Clocks (STPCLK# low) *000* / Default (will be 50%) / 512 *001* / 87.75% / 896 *010* / 75% / 768 011 / 62.5% / 640 100 / 50% / 512 101 / 37.5% / 384 110 / 25% / 256 111 / 12.5% / 128	000	RW
00	Reserved	Reserved.	0	

22.5.3.6 Offset 14h: LV2 – Level 2 Register

Reads to this register return all zeros, writes to this register have no effect. Reads to this register generate a “enter a level 2 power state” (C2) to the clock control logic. This will cause the STPCLK# signal to go active, and stay active until a break event occurs. Throttling (due to THTL_EN or FORCE_THTL) will be ignored.

Note: This register must not be used by systems with more than 1 logical processor, unless appropriate semaphoring software has been put in place to ensure that all threads/ processors are ready for the C2 state when the read to this register occurs.

Table 663. Offset 14h: LV2 – Level 2 Register

I/O Address: PMBASE + 14h (ACPI P_BLK+4) ***Size:*** 8-bit
Default Value: 00h ***Power Well:*** Core
Lockable: No
Usage: ACPI or Legacy

22.5.3.7 Offset 28h: GPE0_STS – General Purpose Event 0 Status Register

Note: This register is symmetrical to the General Purpose Event 0 Enable Register. Unless indicated otherwise below, if the corresponding _EN bit is set, then when the STS bit get set, Intel® 3100 Chipset will generate a Wake Event. Once back in an S0 state (or if already in an S0 state when the event occurs), Intel® 3100 Chipset will also generate an SCI if the SCIEN (PMBASE + 04h, bit 0) bit is set, or an SMI# if the SCIEN bit is not set. Bits 31:16 are reset by a CF9h write; bits 15:0 are not. Bits 31:0 are reset by RSMRST#.

Table 664. Offset 28h: GPE0_STS – General Purpose Event 0 Status Register (Sheet 1 of 3)

I/O Address: *PMBASE + 28h (ACPI GPE0_BLK)* **Size:** *32-bit*
Default Value: *00000000h* **Power Well:** *Resume*
Lockable: *No*
Usage: *ACPI*

Bits	Name	Description	Reset Value	Access
31:16	GPI[n]_STS	0 = Software clears this bit by writing a 1 to it. 1 = These bits are set any time the corresponding GPIO is set up as an input and the corresponding GPIO signal is high (or low if the corresponding GP_INV bit is set). If the corresponding enable bit is set in the GPE0_EN register, then when the GPI[n]_STS bit is set: • If the system is in an S3 or S5 state, the event will also wake the system. • If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPI_ROUT bits for the corresponding GPI.	0	RWC
15:14	Reserved	Reserved	0	
13	PME_B0_STS	***Note:*** This bit will be set to 1 by Intel® 3100 Chipset when any internal device with PCI Power Management capabilities on bus 0 asserts the equivalent of the PME# signal. Additionally, if the PME_B0_EN bit is set, and the system is in an S0 state, then the setting of the PME_B0_STS bit will generate an SCI (or SMI# if SCI_EN is not set). If the PME_B0_STS bit is set, and the system is in an S3 state (or S5 state due to SLP_TYP and SLP_EN), then the setting of the PME_B0_STS bit will generate a wake event, and an SCI (or SMI# if SCI_EN is not set) will be generated. If the system is in an S5 state due to power button override, then the PME_B0_STS bit will not cause a wake event or SCI. The default for this bit is 0. Writing a 1 to this bit position clears this bit	0	RWC
12	Reserved.	Reserved.	0	
11	PME_STS	This bit will be set to 1 by hardware when the PME# signal goes active. [Note Intel® 3100 Chipset might be the cause of PME# going active in some cases]. Additionally, if the PME_EN bit is set, and system is in an S0 state, then the setting of the PME_STS bit will generate an SCI (or SMI# if SCI_EN is not set). If the PME_EN bit is set, and the system is in an S3 state (or S5 state due to SLP_TYP and SLP_EN), then the setting of the PME_STS bit will generate a wake event, and an SCI (or SMI# if SMI_EN is not set) will be generated. If the system is in an S5 state due to power button override, then PME_STS will not cause a wake event or SCI. This bit is cleared by writing a 1 to this bit position.	0	RWC
	Reserved	Reserved.		

Table 664. Offset 28h: GPE0_STS – General Purpose Event 0 Status Register (Sheet 2 of 3)

I/O Address: PMBASE + 28h (ACPI GPE0_BLK) **Size:** 32-bit
Default Value: 00000000h **Power Well:** Resume
Lockable: No
Usage: ACPI

Bits	Name	Description	Reset Value	Access
09	PCI_EXP_STS	0 = Software clears this bit by writing a 1 to it. 1 = Set to 1 by hardware to indicate that: • The PME event message was received on one or more of the PCI Express Ports • An Assert PMEGPE message received from the IMCH via NSI ***Notes:*** 1. The PCI WAKE# pin and the PCI Express Beacons have no impact on this bit. 2. Software attempts to clear this bit by writing a 1 to this bit position. If the PCI_EXP_STS bit went active due to an Assert PMEGPE message, then a Deassert PMEGPE message must be received prior to the software write in order for the bit to be cleared. 3. If the bit is not cleared and the corresponding PCI_EXP_EN bit is set, the level-triggered SCI will remain active. 4. A race condition exists where the PCI Express device sends another PME message because the PCI Express device was not serviced within the time when it must resend the message. This may result in a spurious interrupt, and this is comprehended and approved by the *PCI Express Base Specification*. The window for this race condition is approximately 95-105 ms.	0	RWC
08	RI_STS	0 = Software clears this bit by writing a 1 to it. 1 = Set by hardware when the RI# input signal goes active. The value of this bit is maintained through a G3 state.	0	RWC
07	SMB_WAK_STS	0 = Wake event not caused by the Intel® 3100 Chipset's SMBus logic. 1 = Set by hardware to indicate that the wake event was caused by the Intel® 3100 Chipset's SMBus logic. ***Note:*** 1. If SMB_WAK_STS is set due to SMBus slave receiving a message, it will be cleared by internal logic when CPUTHRMTRIP event happens or by a Power Button Override event. However, CPUTHRMTRIP or Power Button override event will not clear SMB_WAK_STS if it was set due to SMBALERT# signal going active. 2. The SMBus controller will independently cause an SMI# so this bit does not need to do so (unlike the other bits in this register). 3. This bit is set by the SMBus slave command 01h (Wake/SMI#) even when the system is in the S0 state. Therefore, to avoid an instant wake on subsequent transitions to sleep states, software must clear this bit after each reception of the Wake/SMI# command or just prior to entering the sleep state. 4. The SMBALERT_STS bit (D31:F3:I/O Offset 00h:bit 5) must be cleared by software before clearing this bit.	0	RWC
06	TCOSCI_STS	0 = TCO logic did Not cause SCI. 1 = Set by hardware when the TCO logic causes an SCI. This bit can be reset by writing a one to this bit position.	0	RWC
05	Reserved	Reserved	0	

Table 664. Offset 28h: GPE0_STS – General Purpose Event 0 Status Register (Sheet 3 of 3)

I/O Address: PMBASE + 28h (ACPI GPE0_BLK) **Size:** 32-bit
Default Value: 00000000h **Power Well:** Resume
Lockable: No
Usage: ACPI

Bits	Name	Description	Reset Value	Access
04	USB2_STS	0 = USB UHCI controller 2 does Not need to cause a wake. 1 = Set by hardware when USB UHCI controller 2 needs to cause a wake. Wake event will be generated if the corresponding USB2_EN bit is set. This bit is only set by hardware and can be reset by writing a one to this bit position or a resume-well reset.	0	RWC
03	USB1_STS	0 = USB UHCI controller 1 does Not need to cause a wake. 1 = Set by hardware when USB UHCI controller 1 needs to cause a wake. Wake event will be generated if the corresponding USB1_EN bit is set. This bit is only set by hardware and can be reset by writing a one to this bit position or a resume-well reset.	0	RWC
02	Reserved	Reserved.	0	
01	HOT_PLUG_STS	0 = This bit is cleared by writing a 1 to this bit position. 1 = When a PCI Express hot-plug event occurs. This will cause an SCI if the HOT_PLUG_EN bit is set in the GEP0_EN register.	0	RWC
00	THRM_STS	0 = THRM# signal Not driven active as defined by the THRM_POL bit 1 = Set by hardware anytime the THRM# signal is driven active as defined by the THRM_POL bit. Additionally, if the THRM_EN bit is set, then the setting of the THRM_STS bit will also generate a power management event (SCI or SMI#). This bit is cleared by S/W writing a one to this bit position or a resume-well reset.	0	RWC

22.5.3.8 PMBASE Offset 2Ch: GPE0_EN – General Purpose Event 0 Enables Register

Note: This register is symmetrical to the General Purpose Event 0 Status Register. All the bits in this register must be cleared to 0 based on a Power Button Override-CPU Thermal Trip event, or internal thermal sensor catastrophic condition. Unless otherwise noted, all bits are in the resume well. The Resume well bits are all cleared by RSMRST# and RTC well bits are cleared by RTEST#.

Table 665. PMBASE Offset 2Ch: GPE0_EN – General Purpose Event 0 Enables Register (Sheet 1 of 2)

I/O Address: PMBASE + 2Ch (ACPI GPE0_BLK +4) **Size:** 32-bit

Default Value: 00000000h **Power Well:** Bits 0-7 Resume, Bits 8, 10-11, 13 RTC, Bits 9, 12, 16:31 Resume

Lockable: No

Usage: ACPI

Bits	Name	Description	Reset Value	Access
31:16	GPI[n]_EN:	These bits enable the corresponding GPI[n]_STS bits being set to cause an SCI and/or wake event. These bits are cleared by RSMRST#. ***Note:*** Mapping is as follows: bit 31 corresponds to GPI[15]... and bit 16 corresponds to GPI:[0]. Software should not set bits 28:29 corresponding to GPI[12:13].	0	RW
15	Reserved	Reserved.	0	
14	Reserved	Reserved.	0	
13	PME_B0_EN	0 = Disable 1 = Enables the setting of the PME_B0_STS bit to generate a wake event and/or an SCI or SMI#. PME_B0_STS can be a wake event from the S3 state, or from S5 (if entered via SLP_TYP and SLP_EN) or power failure, but not Power Button Override. This bit defaults to 0. ***Note:*** It is only cleared by Software or RTEST#. It is not cleared by CF9h writes. This bit is in the RTC well.	0	RW
12	Reserved	Reserved.	0	
11	PME_EN	0 = Disable. 1 = Enables the setting of the PME_STS to generate a wake event and/or an SCI. PME# can be a wake event from the S3 state or from S5 (if entered via SLP_EN, but not power button override). This bit is only cleared by software or RTEST#. It is not cleared by CF9h writes. This bit is in the RTC well.	0	RW
10	Reserved	Reserved		
09	PCI_EXP_EN	0 = Disable SCI generation upon PCI_EXP_STS bit being set. 1 = Enables Intel® 3100 Chipset to cause an SCI when PCI_EXP_STS bit is set. This is used to allow the PCI Express ports, including the link to the IMCH, to cause an SCI due to wake/PME events.	0	RW
08	RI_EN	When RI_EN and RI_STS are both set, a Wake event will occur. If RI_EN is not set, then when RI_STS is set, no Wake event will occur. 0 = Disable. 1 = Enables the setting of the RI_STS to generate a wake event. This bit is only cleared by software or RTEST#. This bit is in the RTC well.	0	RW
07	Reserved	Reserved.	0	
06	TCOSCI_EN	When TCOSCI_EN and TCOSCI_STS are both set, an SCI will be generated. 0 = Disable. 1 = Enables the setting of the TCOSCI_STS to generate an SCI. This bit is in the resume well. This bit is only cleared by software or RSMRST#. It is not cleared by CF9h writes.	0	RW
05	Reserved	Reserved	0	
04	USB2_EN	0 = Disable. 1 = Enables the setting of the USB2_STS to generate a wake event. The USB2_STS bit is set anytime USB controller 2 signals a wake event. Break events are handled via the USB interrupt.	0	RW
03	USB1_EN	0 = Disable. 1 = Enables the setting of the USB1_STS to generate a wake event. The USB1_STS bit is set anytime USB controller 1 signals a wake event. Break events are handled via the USB interrupt.	0	RW

Table 665. PMBASE Offset 2Ch: GPE0_EN – General Purpose Event 0 Enables Register (Sheet 2 of 2)

I/O Address: PMBASE + 2Ch (ACPI GPE0_BLK +4) **Size:** 32-bit
Default Value: 00000000h **Power Well:** Bits 0-7 Resume, Bits 8, 10-11, 13 RTC, Bits 9, 12, 16:31 Resume
Lockable: No
Usage: ACPI

Bits	Name	Description	Reset Value	Access
02	THRM_POL	This bit controls the polarity of the THRM# pin needed to set the THRM_STS bit. 0 = Low value on the THRM# signal will set the THRM_STS bit. 1 = HIGH value on the THRM# signal will set the THRM_STS bit.	0	RW
01	HOT_PLUG_EN	0 = Disables SCI generation upon the HOT_PLUG_STS bit being set. 1 = Enables Intel® 3100 Chipset to cause an SCI when the HOT_PLUG_STS bit is set. This is used to allow the PCI Express ports to cause an SCI due to hot-plug events.	0	RW
00	THRM_EN	0 = Disable. 1 = Active assertion of the THRM# signal (as defined by the THRM_POL bit) will set the THRM_STS bit and generate a power management event (SCI or SMI).	0	RW

22.5.3.9 Offset 30h: SMI_EN – SMI Control and Enable Register

Note: This register is symmetrical to the SMI Status Register.

Table 666. Offset 30h: SMI_EN – SMI Control and Enable Register (Sheet 1 of 2)

I/O Address: PMBASE + 30h **Size:** 32-bit
Default Value: 00000000h **Power Well:** Core
Lockable: No
Usage: ACPI or Legacy

Bits	Name	Description	Reset Value	Access
31:19	Reserved	Reserved	0	
18	INTEL_ USB2_EN	0 = Disables Intel-Specific USB 2.0 SMI logic. 1 = Enables Intel-Specific USB 2.0 SMI logic to cause SMI#.	0	RW
17	LEGACY_ USB2_EN	0 = Disable 1 = Enables legacy USB 2.0 logic to cause SMI#.	0	RW
16:15	Reserved	Reserved	0	
14	PERIODIC_ EN	0 = Disable 1 = Enables the Intel® 3100 Chipset to generate an SMI# when the PERIODIC_STS bit (PMBASE + 34h, bit 14) is set in the SMI_STS register (PMBASE + 34h).	0	RW
13	TCO_EN	0 = Disables TCO logic generating an SMI#. 1 = Enables the TCO logic to generate SMI#. See Chapter 27.0 for more details on TCO functions. If the NMI2SMI_EN bit is set, then SMIs that are caused by NMIs (i.e., rerouted) will not be gated by the TCO_EN bit. Even if the TCO_EN bit is 0, the NMIs will still be routed to cause the SMI#. ***Note:*** This bit can not be written once the TCO_LOCK bit (at offset 08h of TCO I/O Space) is set. This prevents unauthorized software from disabling the generation of TCO-based SMIs.	0	RW
12	Reserved	Reserved	0	

Table 666. Offset 30h: SMI_EN – SMI Control and Enable Register (Sheet 2 of 2)

I/O Address: PMBASE + 30h
Size: 32-bit
Default Value: 00000000h
Power Well: Core
Lockable: No
Usage: ACPI or Legacy

Bits	Name	Description	Reset Value	Access
11	MCSMI_EN	0 = Disable. 1 = Enables IICH to trap accesses to the microcontroller range (62h or 66h) and generate an SMI#. Note that "trapped' cycles will be claimed by the IICH on PCI, but not forwarded to LPC.	0	RW
10:08	Reserved	Reserved	0	
07	BIOS_RLS	0 = This bit will always return 0 on reads. Writes of 0 to this bit have no effect. 1 = Enables the generation of an SCI interrupt for ACPI software when a one is written to this bit position by BIOS software. ***Note:*** GBL_STS being set will cause an SCI, even if the SCI_EN bit is not set. Software must take great care not to set the BIOS_RLS bit (which causes GBL_STS to be set) if the SCI handler is not in place.	0	WO
06	SWSMI_ TMR_EN	0 = Disable. Clearing the SWSMI_TMR_EN bit before the timer expires will reset the timer and the SMI# will not be generated. 1 = Starts Software SMI# Timer. When the SWSMI timer expires (the timeout period depends upon the SWSMI_RATE_SEL bit setting), SWSMI_TMR_STS is set and an SMI# is generated. SWSMI_TMR_EN stays set until cleared by software.	0	RW
05	APMC_EN	0 = Writes to the APM_CNT register will not cause an SMI#. 1 = Enables writes to the APM_CNT register to cause an SMI#	0	RW
04	SMI_ON_ SLP_EN	0 = Disables the generation of SMI# on SLP_EN. Note that this bit must be 0 before the software attempts to transition the system into a sleep state by writing a 1 to the SLP_EN bit. 1 = A write of 1 to the SLP_EN bit (bit 13 in PM1_CNT register) will generate an SMI#, and the system will not transition to the sleep state based on that write to the SLP_EN bit. This allows the SMI# handler work around chip-level bugs. It is expected that the SMI# handler will turn off the SMI_ON_SLP_EN bit before actually setting the SLP_EN bit.	0	RW
03	LEGACY_ USB_EN	0 = Disables legacy USB circuit 1 = Enables legacy USB circuit to cause SMI#.	0	RW
02	BIOS_EN	0 = Disables the generation of SMI# when ACPI software writes a 1 to the GBL_RLS bit. 1 = Enables the generation of SMI# when ACPI software writes a 1 to the GBL_RLS bit.	0	RW
01	EOS	End of SMI. This bit controls the arbitration of the SMI signal to the processor. This bit must be set in order for Intel® 3100 Chipset to assert SMI# low to the processor after SMI# has been asserted previously. 0 = Once Intel® 3100 Chipset asserts SMI# low, the EOS bit is automatically cleared. 1 = In the SMI handler, the processor must clear all pending SMIs (by servicing them and then clearing their respective status bits), set the EOS bit, and exit SMM. This will allow the SMI arbiter to reassert SMI upon detection of an SMI event and the setting of a SMI status bit. The SMI# signal will go inactive for 4 PCI clocks.	0	RW
00	GBL_SMI_EN	0 = No SMI# will be generated. 1 = Enables the generation of SMIs in the system upon any enabled SMI event. This bit is reset by a PCI reset event. ***Note:*** When the SMI_LOCK bit is set, this bit cannot be changed.	0	RW

22.5.3.10 Offset 34h: SMI_STS – SMI Status Register

Note: If the corresponding _EN bit is set when the _STS bit is set, Intel® 3100 Chipset will cause an SMI# (except bits 8-10, which do not cause SMI#.)

Table 667. Offset 34h: SMI_STS – SMI Status Register (Sheet 1 of 3)

I/O Address: PMBASE + 34h
Size: 32 bit
Default Value: 00000000h
Power Well: Core
Lockable: No
Usage: ACPI or Legacy

Bits	Name	Description	Reset Value	Access
31:21	Reserved	Reserved	0	
20	PCI_EXP_ SMI_ST	0 = PCI Express SMI event did not occur. 1 = PCI Express SMI event occurred. This could be due to a PCI Express PME event or Hot Plug Event. See the Power Management sub-section in the PCI Express Operation section under the PCI Express Chapter in Volume 4 for more details on how this bit is set and cleared.	0	RO
19	Reserved	Reserved	0	
18	INTEL_ USB2_STS	This non-sticky read-only bit is a logical OR of each of the SMI status bits in the Intel-Specific USB 2.0 SMI Status Register ANDed with the corresponding enable bits. This bit will not be active if the enable bits are not set. Writes to this bit will have no effect.	0	RO
17	LEGACY_ USB2_STS	This non-sticky read-only bit is a logical OR of each of the SMI status bits in the USB 2.0 Legacy Support Register ANDed with the corresponding enable bits. This bit will not be active if the enable bits are not set. Writes to this bit will have no effect.	0	RO
16	SMBUS_ SMI_STS	Intel® 3100 Chipset sets this bit to 1 to indicate that the SMI# was caused by: A. The SMBus Slave receiving a message that an SMI# must be caused. B. The SMBALERT# signal goes active and the SMB_SMI_EN bit is set and the SMBALERT_DIS bit is cleared. C. The SMBus Slave receiving a HOST_NOTIFY message and the HOST_NOTIFY_INTREN and the SMB_SMI_EN bits are set. D. The SMBus Slave receiving a "SMI in S0" message. This bit is sticky. It is cleared by writing a 1 to this bit position. ***Note:*** This bit is set from the 64 KHz clock domain used by the SMBus. Software must wait at least 15.63 µs (= 1/64 kHz) after the initial assertion of this bit before clearing it.	0	RWC
15	SERIRQ_ SMI_STS	0 = SMI# not caused by SERIRQ decoder. 1 = Indicates the SMI# was caused by the SERIRQ decoder. ***Note:*** This bit is **not** sticky. Writes to this bit will have no effect.	0	RO
14	PERIODIC_STS	This bit will be set at the rate determined by the PER_SMI_SEL bits. If the PERIODIC_EN bit is also set, tIntel® 3100 Chipset will generate an SMI#. This bit is cleared by writing a 1 to this bit position.	0	RWC
13	TCO_STS	0 = SMI not caused by TCO logic. 1 = Indicates SMI was caused by the TCO logic. ***Note:*** Will not cause wake event. This bit is cleared by writing a 1 to this bit position.	0	RWC
12	DEVMON_STS	This read-only bit is set when bit 0 in the DEVTRAP_STS register is set. It is not sticky, so writes to this bit will have no effect. See Section 22.5.1.3.	0	RO

Table 667. Offset 34h: SMI_STS – SMI Status Register (Sheet 2 of 3)

I/O Address: PMBASE + 34h
Size: 32 bit
Default Value: 00000000h
Power Well: Core
Lockable: No
Usage: ACPI or Legacy

Bits	Name	Description	Reset Value	Access
11	MCSMI_STS	0 = Indicates that there has been no access to the power management microcontroller range (62h or 66h). 1 = Set if there has been an access to the power management microcontroller range (62h or 66h) and the Microcontroller Decode Enable #1 bit in the LPC Bridge I/O Enables configuration register is 1. Note that this implementation assumes that the Microcontroller is on LPC, if this changes in the future (i.e. PCI e-based SIO), then the implementation will need to remove the LPC Decode Enable dependency. If this bit is set, and the MCSMI_EN bit is also set, Intel® 3100 Chipset will generate an SMI#. This bit is set by hardware and cleared by software writing a 1 to its bit position.	0	RWC
10	GPE1_STS	This bit is a logical OR of the bits in the ALT_GPI_SMI_STS register that are also set up to cause an SMI# (as indicated by the GPI_ROUT registers) and have the corresponding bit set in the ALT_GPI_SMI_EN register. 0 = SMI# was not generated by a GPI assertion. 1 = SMI# was generated by a GPI assertion. Bits that are not routed to cause an SMI# will have no effect on this bit. This bit is NOT sticky. Writes to this bit will have no effect.	0	RO
09	GPE0_STS	This bit is a logical OR of bits 13, 11, 8:6, 4:3 and 0 in the GPE0_STS register (PMBASE + 28h) that also have the corresponding bit set in the GPE0_EN register (PMBASE + 2Ch). This bit is NOT sticky. 0 = SMI# was not generated by a GPE0 event. 1 = SMI# was generated by a GPE0 event. ***Note:*** Writes to this bit will have no effect. The setting of this bit does not cause the SMI# ***Note:*** Bits 31:16 of the GPE0_STS register are not capable of generating SMIs; therefore, they do not set this SMI status bit.	0	RO
08	PM1_STS_REG	This is an OR of the bits (except for bits 5 and 4) in the ACPI PM1 Status Reg. (offset PMBASE+00h). Not sticky. Writes to this bit have no effect. 0 = SMI# was not generated by a PM1_STS event. 1 = SMI# was generated by a PM1_STS event. ***Note:*** The setting of this bit does not cause the SMI#.	0	RO
07	Reserved	Reserved.	0	
06	SWSMI_ TMR_STS	0 = Software SMI# Timer has Not expired. 1 = Set by the hardware when the Software SMI# Timer expires. This bit will remain 1 until the software writes a 1 to this bit.	0	RWC
05	APM_STS	SMI# was generated by a write access to the APM control register and if the APMC_EN bit is set. 0 = No SMI# generated by write access to APM Control register with APMCH_EN bit set. 1 = SMI# was generated by a write access to the APM Control register with the APMC_EN bit set. This bit is cleared by writing a one to its bit position.	0	RWC

Table 667. Offset 34h: SMI_STS – SMI Status Register (Sheet 3 of 3)

I/O Address: PMBASE + 34h
Size: 32 bit
Default Value: 00000000h
Power Well: Core
Lockable: No
Usage: ACPI or Legacy

Bits	Name	Description	Reset Value	Access
04	SMI_ON_ SLP_EN_STS	This bit will be set by Intel® 3100 Chipset when a write access attempts to set the SLP_EN bit. 0 = No SMI# caused by write of 1 to SLP_EN bit when SLP_SMI_EN bit is also set. 1 = Indicates an SMI# was caused by a write of 1 to SLP_EN bit when SLP_SMI_EN bit is also set. This bit is cleared by writing a 1 to this bit position.	0	RWC
03	LEGACY_ USB_STS	This non-sticky read-only bit is a logical OR of each of the SMI status bits in the USB Legacy Keybd Register ANDed with the corresponding enable bits. This bit will not be active if the enable bits are not set. 0 = SMI# was not generated by USB Legacy event. 1 = SMI# was generated by USB Legacy event.	0	RO
02	BIOS_STS	0 = No SMI# generated due to ACPI software requesting attention. 1 = SMI# was generated due to ACPI software requesting attention (writing a 1 to the GBL_RLS bit with the BIOS_EN bit set).	0	RWC
01:00	Reserved	Reserved.	0	

22.5.3.11 Offset 38h: ALT_GPI_SMI_EN – Alternate GPI SMI Enable Register

Table 668. Offset 38h: ALT_GPI_SMI_EN – Alternate GPI SMI Enable Register

I/O Address: 38h
Size: *16-bit*
Default Value: 0000h
Power Well: Resume
Lockable: No
Usage: ACPI or Legacy

Bits	Name	Description	Reset Value	Access
15:00	ALT_GPI_ SMI_EN	These bits are used to enable the corresponding GPIO to cause an SMI#. In order for these bits to have any effect, the following must be true. 1. The corresponding bit in the ALT_GPI_SMI_EN register is set. 2. The corresponding GPI must be routed in the GPI_ROUT register to cause an SMI. 3. The corresponding GPIO must be implemented. All bits are in the resume well.	0000h	RW

22.5.3.12 Offset 3Ah: ALT_GPI_SMI_STS – Alternate GPI SMI Status Register

Table 669. Offset 3Ah: ALT_GPI_SMI_STS – Alternate GPI SMI Status Register

I/O Address: 3Ah
Size: Read/Write-Clear
Default Value: 0000h
Power Well: Resume
Lockable: No
Usage: ACPI or Legacy

Bits	Name	Description	Reset Value	Access
15:00	ALT_GPI_ SMI_STS	These bits report the status of the corresponding GPIs. 1 = active, -0 = inactive. These bits are sticky. If the following conditions are true, then an SMI# will be generated and the ALT_GPI_SMI_STS bit set: 1. The corresponding bit in the ALT_GPI_SMI_EN register is set 2. The corresponding GPI must be routed in the GPI_ROUT register to cause an SMI. 3. The corresponding GPIO must be implemented. All bits are in the resume well. Default for these bits are dependent on the state of the GPI pins.	0000h	RWC

22.5.3.13 Offset 44h: DEVTRAP_STS Register

Each bit indicates if an access has occurred to the corresponding devices trap range, or for bits 6:9 if the corresponding PCI interrupt is active. Write 1 to the same bit position to clear it. This register is used by APM power management software to see if there has been system activity. The periodic SMI# timer indicates if it is the right time to read the DEVTRAP_STS register (PMBASE + 44h).

Table 670. Offset 44h: DEVTRAP_STS Register (Sheet 1 of 2)

I/O Address: *PMBASE +44h*
Size: *16-bit*
Default Value: *0000h*
Power Well: *Core*
Lockable: *No*
Usage: *Legacy Only*

Bits	Name	Description	Reset Value	Access
15:13	Reserved	Reserved	0	
12	D12_TRP_STS	KBC (60/64h): 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.	0	RWC
11:10	Reserved	Reserved	0	
09	D9_TRP_STS	PIRQ[D or H]: 0 = The corresponding PCI interrupts have not been active. 1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location.	0	RWC
08	D8_TRP_STS	PIRQ[C or G]: 0 = The corresponding PCI interrupts have not been active. 1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location.	0	RWC

Table 670. Offset 44h: DEVTRAP_STS Register (Sheet 2 of 2)

I/O Address: PMBASE +44h	**Size:** 16-bit
Default Value: 0000h	**Power Well:** Core
Lockable: No	
Usage: Legacy Only	

Bits	Name	Description	Reset Value	Access
07	D7_TRP_STS	PIRQ[B or F]: 0 = The corresponding PCI interrupts have not been active. 1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location.	0	RWC
06	D6_TRP_STS	PIRQ[A or E]: This bit will be set if PCI IRQ A or PCI IRQ E goes active (by the pin or internal signal). 0 = The corresponding PCI interrupts have not been active. 1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location.	0	RWC
05:00	Reserved	Reserved	0	

22.6 SMI#/SCI Generation

Upon any SMI# event taking place, IICH will assert SMI# to the processor, which will cause it to enter SMM space. SMI# remains active until the EOS bit is set. When the EOS bit is set, SMI# will go inactive for a minimum of 4 PCICLK. If another SMI event occurs, SMI# will be driven active again.

The SCI is a level-mode interrupt that is typically handled by an ACPI-aware operating system. In non-APIC systems (which is the default), the SCI IRQ is routed to one of the 8259 interrupts (IRQ9, 10, or 11). The 8259 interrupt controller must be programmed to level mode for that interrupt.

In systems using the APIC, the SCI can be routed to interrupts 9, 10, 11, 20, 21, 22, or 23. The interrupt polarity changes depending on whether it is on an interrupt shareable with a PIRQ or not; see Section 22.5.1 for details. The interrupt will remain asserted until all SCI sources are removed.

Table 671 shows which events can cause an SCI and Table 673 shows the causes of an SMI#.

Note: Some events can be programmed to cause either an SMI# or SCI. The usage of the event for SCI (instead of SMI#) is typically associated with an ACPI-based system.

Table 671. Causes of SCI

Cause	Additional Enables (See Note 1)	Where Reported
PME#	PME_EN = 1	PME_STS
Internal EHCI wake (PME_B0)	PME_B0_EN = 1	PME_B0_STS
PCI Express PME messages	PCI_EXP_EN = 1 and (Not enabled for SMI)	PCI_EXP_STS
PCI Express Hot Plug Message	HOT_PLUG_EN = 1 and (Not enabled for SMI)	HOT_PLUG_STS
Power Button Press	PWRBTN_EN = 1	PWRBTN_STS
RTC Alarm	RTC_EN = 1	RTC_STS

Table 671. Causes of SCI

Ring Indicate	RI_EN = 1	RI_STS
USB #1 wakes	USB1_EN = 1	USB1_STS
USB #2 wakes	USB2_EN = 1	USB2_STS
THRM# pin active (based on THRM_POL)	THRM_EN = 1	THRM_STS
ACPI Timer overflow (2.34 seconds)	TMROF_EN = 1	TMROF_STS
Any GPI	GPI[X]_Route = 10, GPE0[X]_EN = 1	GPE0[X]_STS
TCO SCI Logic (see Table 672)	TCOSCI_EN = 1	TCOSCI_STS
BIOS_RLS written to 1	GBL_EN = 1	GBL_STS

Notes: Causes of SCI:
1. SCI_EN must be 1 to enable SCI
2. SCI can be routed to cause Interrupt 9:11 or 20:23 (20:23 only available in APIC mode)

Note: There are various sources that cause the TCO SCI, shown in Table 672.

Table 672. Causes of TCO SCI

Cause	Additional Enables	Where Reported
Message from IMCH	None	IMCHSCI_STS

See Chapter 27.0 for more details on the causes of the TCO SCI.

Table 673. Causes of SMI# (Sheet 1 of 2)

Cause	Additional Enables	Where Reported	Synch
PME#	SCI_EN = 0, PME_EN = 1	PME_STS	
Internal EHCI wake (PME_B0)	SCI_EN = 0, PME_B0_EN = 1	PME_B0_STS	
PCI Express PME messages	See PCI Express Section in Volume 4.	PCI_EXP_SMI_STS	
PCI Express Hot Plug Message	See PCI Express Section in Volume 4.	PCI_EXP_SMI_STS	
Power Button Press	SCI_EN = 0, PWRBTN_EN = 1	PWRBTN_STS	
RTC Alarm	SCI_EN = 0, RTC_EN = 1	RTC_STS	
Ring Indicate	SCI_EN = 0, RI_EN = 1	RI_STS	
USB #1 wakes	SCI_EN = 0, USB1_EN = 1	USB1_STS	
USB #2 wakes	SCI_EN = 0, USB2_EN = 1	USB2_STS	
THRM# pin active (based on THRM#_ POL)	SCI_EN = 0, THRM_EN = 1	THRM_STS	
ACPI Timer overflow (2.34 seconds)	SCI_EN = 0, TMROF_EN = 1	TMROF_STS	
Any GPI	GPI[X]_Route = 01, ALT_GPI_SMI[X]_EN = 1	GPE1_STS, ALT_GPI_SMI[X]_STS	
TCO SMI Logic (see Table 666)	TCO_EN = 1	TCO_STS	
NMI I (and NMIs mapped to SMI) See NMI section for causes.	NMI2SMI_EN = 1	TCO_STS, NMI2SMI_STS	
GBL_RLS written to 1	BIOS_EN = 1	BIOS_STS	X
Write to B2h register	APMC_EN	APM_STS	X
Periodic timer expires	PERIODIC_EN = 1	PERIODIC_STS	
64 ms timer expires	SWSMI_TMR_EN = 1	SWSMI_TMR_STS	

Table 673. Causes of SMI# (Sheet 2 of 2)

Cause	Additional Enables	Where Reported	Synch
Enhanced USB Legacy Support Event	LEGACY_USB2_EN = 1	LEGACY_USB2_STS	
Enhanced USB Intel-Specific Event	INTEL_USB2_EN = 1	INTEL_USB2_STS	
Classic USB Legacy logic (Port 64/60 rd/wr, End of pass-through)	LEGACY_USB_EN = 1	LEGACY_USB_STS	X
Classic USB Legacy logic (IRQ)	LEGACY_USB_EN = 1	LEGACY_USB_STS	
Serial IRQ SMI Reported	None	SERIRQ_SMI_STS	
Device Monitors (D15:D0)matches an address in its range	See Trap Section 23.2.2.1.	DEVMON_STS,	X
SMBus Host Controller	SMB_SMI_EN, Host Controller enabled	Various bits in the SMBus Host Status Register	
SMBus Slave SMI message	None	SMBUS_SMI_STS	
SMBus SMBALERT# signal active	None	SMBUS_SMI_STS	
SMBus Host Notify message received	HOST_NOTIFY_INTREN	SMBUS_SMI_STS, HOST_NOTIFY_STS	
Access to Microcontroller Range (62h/66h)	MCSMI_EN	MCSMI_STS	X
SLP_EN bit written to 1	SMI_ON_SLP_EN = 1	SMI_ON_SLP_EN_STS	X

Notes: Causes of SMI#:
1. GBL_SMI_EN must be 1 to enable SMI
2. EOS must be written to 1 to reenable SMI for the next one
3. Some SMI#s are considered “synchronous”, in that the processor must recognize the SMI# prior to completing the instruction (I/O read, I/O write, Memory read, or Memory write) that must cause the SMI#. This is accomplished by having the SMI# signal go active to the processor prior to the processor observing the RDY# signal that terminates the cycle. SMI#s marked with X in the Synch column are treated as Synchronous. Synchronous SMI#s are not possible in IA-64 platforms, since they do not support the SMI# signal.
4. NMI2SMI_STS isn’t gated by TCO_EN.
5. Intel® 3100 Chipset must have SMI# fully enabled when Intel® 3100 Chipset is also enabled to trap cycles. If SMI# is not enabled in conjunction with the trap enabling, then hardware behavior is undefined.

Table 674. Causes of TCO SMI#

Cause	Additional Enables	Where Reported
Year 2000 Rollover	None	NEWCENTURY_STS
TCO TIMEROUT	None	TIMEOUT
OS writes to TCO_DAT_IN register	None	OS_TCO_SMI
NMI occurred (and NMIs mapped to SMI)	NMI2SMI_EN = 1	NMI2SMI_STS
Note: NMI2SMI_STS isn’t gated by TCO_EN. See Table 667.		
INTRUDER# signal goes active	INTRD_SEL = 10	INTRD_DET
Changes of the BIOSWP bit from 0 to 1	BLD = 1	BIOSWR_STS
Message from IMCH		IMCHSMI_STS
Write attempted to BIOS	BIOSWP = 1	BIOSWR_STS

See Chapter 27.0 for details on the TCO SMI# causes.

22.6.0.1 PCI Express* SCI

PCI Express ports and the IMCH (via NSI) have the ability to cause PME using messages. When a PME message is received, Intel® 3100 Chipset will set the PCI_EXP_STS bit. If the PCI_EXP_EN bit is also set, Intel® 3100 Chipset can cause an SCI via the GPE1_STS register.

22.7 Dynamic Processor Clock Control

22.7.1 Overview

Intel® 3100 Chipset has primary control for dynamically starting and stopping system clocks. The clock control is used for the transitions among the various S0/Cx states (i.e., processor throttling). Each dynamic clock control method is described in this section. The various Sleep states may also perform types of non-dynamic clock control, and are described in Section 22.6.

Intel® 3100 Chipset supports the ACPI C0, C1and C2 states

The Dynamic Clock control is handled using the following signals:

- STPCLK# ..... Used to halt processor instruction stream

The C1 state is entered based on the processor performing an autohalt instruction.

The C2 state is entered based on the processor reading the Level 2 register.

The C1 and C2 states end due to a Break event. Based on the break event, Intel® 3100 Chipset returns the system to a C0 state. Table 675 lists the possible break events from the C2 states. The break events from the C1 state are indicated in the processor's datasheet.

Table 675. Break Events

Event	Breaks From	Comment
Any unmasked interrupt goes active	C2	IRQ[0:15] when using the 8259s, IRQ[0:23] for I/O (X) APIC. Since SCI is an interrupt, any SCI will also be a break event.
Any internal event that will cause an NMI or SMI#	C2	Many possible sources
Any internal event that will cause INIT# to go active	C2	Could be indicated by the keyboard controller via the RCIN input signal.
RTC Interrupt Pending	C2	Only available if the RTC Interrupt (IRQ8) is enabled as a break event (See RTC Interrupt Break Enable bit in Section 21.3.1.1).
Processor Pending Break Event Indication	C2	Only available if FERR# is enabled for break event indication (See FERR# MUX Enable bit in Section 21.3.1.1)
REQ-C0 message from IMCH	C2	Can be sent at any time after the Ack-C2 message and before the Ack-C0 message (i.e., any time not in C0 state).

The Pending Break Event (PBE) indication from the processor is supported using the FERR# signal. The following rules apply:

1. When STPCLK# is detected active by the processor, the FERR# signal from the processor will be redefined to indicate whether an interrupt is pending. The signal is active low (i.e., FERR# will be low to indicate a pending interrupt).

2. When the STPCLK# asserts, it will latch the current state of the FERR# signal and continue to present this state to the FERR# state machine (independent of what the FERR# pin does after the latching).
3. When the Stop-Grant cycle is detected, it will start looking at the FERR# signal as a break event indication. If FERR# is sampled low, a break event is indicated. This will force a transition to the C0 state.
4. When the processor detects the deassertion of STPCLK#, the processor will start driving the FERR# signal with the natural value (i.e., the value it would do if the pin was not muxed). The time from STPCLK# inactive to the FERR# signal transitioning back to the native function must be less than 120 ns.
5. At least 180 ns passes after deasserting STPCLK#. Then it starts using the FERR# signal for an indication of a floating point error. The maximum time that may pass is bounded such that it must have a chance to look at the FERR# signal before reasserting STPCLK#. Based on current implementation, that maximum time would be 240 ns (8 PCI clocks). Since the processor has 120 ns to revert to the proper FERR# function, there are 60 ns of margin inherent in the timings.

The break event associated with this mechanism does not need to set any particular status bit, since the pending interrupt will be serviced by the processor after returning to the C0 state.

22.7.2 Transition Rules Among S0/Cx and Sx States

The following priority rules and assumptions apply among the various S0/Cx and throttling states:

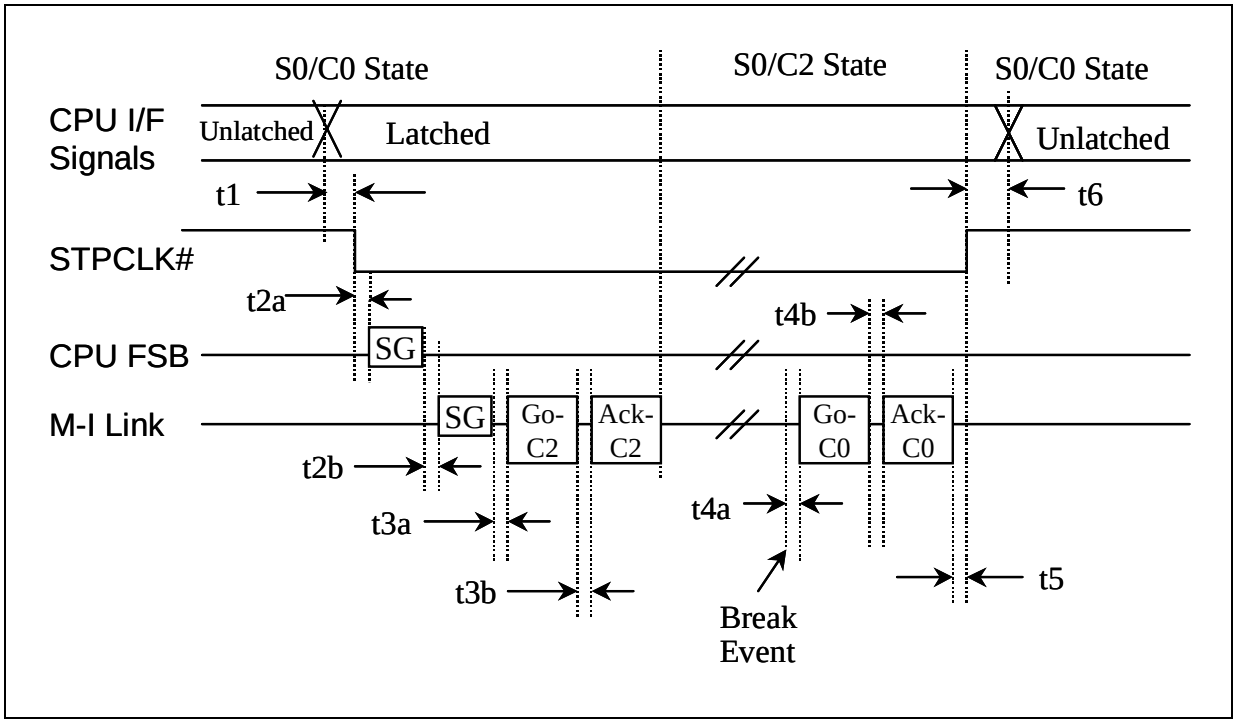
- Entry to any S0/Cx state is mutually exclusive with entry to S3 or S5 state. This is because the processor can only perform one register access at a time and Sleep states have higher priority than thermal throttling.
- When the SLP_EN bit is set (system going to a S3, S5 sleep state), the THTL_EN and FORCE_THTL bits can be internally treated as being disabled (no throttling while going to sleep state).
- If the THTL_EN or FORCE_THTL bits are set, and a Level 2 read then occurs, the system must immediately go and stay in a C2 state until a break event occurs. A Level 2 read has higher priority than the software initiated throttling.
- After an exit from a C2 state (due to a Break event), and if the THTL_EN or FORCE_THTL bits are still set, the system will continue to throttle STPCLK#. The first transition on STPCLK# active can be delayed by up to one THRM period (1024 PCI clocks = 30.72 µs), depending on the time of the break event.
- The IMCH (or equivalent) must post Stop-Grant cycles in such a way that the processor gets an indication of the end of the special cycle prior to Intel® 3100 Chipset observing the Stop-Grant cycle. This ensures that the STPCLK# signal stays active for a sufficient period after the processor observes the response phase.
- If in the C1 state and the STPCLK# signal goes active, the processor will generate a Stop-Grant cycle, and the system must go to the C2-like state. When STPCLK# goes inactive, it must return to the C1 state.

22.7.3 S0/C0, S0/C2, Entry/Exit Timings and Sequences

The timings associated with the C0-C2-C0, sequences are shown in the following figures and tables.

22.7.3.1 C0→C2→C0 Timings and Diagram

Figure 82. C0→C2→C0 Entry/Exit Timings



Note: In Figure 82, the M-I Link must be labeled as "NSI" and the "SG" message on NSI must be "Req-C2"

22.7.3.2 C0→C2 Entry Sequence

Table 676. C0→C2→C0 Timings

Sym	Min	Max	Units	Description
T1	0	Note 1		Processor Interface Signals Latched prior to STPCLK# active. Note that this does not apply for synchronous SMI's.
T2a	0	Note 2		STPCLK# active to Stop-Grant cycle on processor front-side bus (can wait forever)
T2b	0	Note 3		Stop-Grant on FSB to Stop-Grant on NSI. ***Note:*** This is according to an IMCH specification.
T3a	0	Note 1		Stop-Grant on NSI to Go-C2 message. This must be as short as feasible.
T3b	128	Note 3	BCLK	Go C-2 message to Ack-C2 message. Note that this is according to an IMCH specification and is only required if the IMCH has the CPULSLP# signal. It is needed to enforce the Stop-Grant to CPUSLP# timing requirements. If the IMCH does not have the CPUSLP# signal, then this can be 0.
T4a	0	8	PCI CLK	Break Event to when GO_C0 message is ready to be sent. The actual message may be delayed if NSI is busy with other traffic.
T4b	0	Note 3		Go-C0 message to Ack-C0 message. Note that this is an IMCH specification. This must be as short as feasible.
T5	0	Note 1		End of Ack_C0 message to STPCLK# high
T6	8	9	PCI CLK	STPCLK# high to processor interface signals unlatched.

1. This value must be small (a few PCI clocks). For messages, it may be difficult to determine the maximum time, since power management messages may have to wait for other traffic on NSI.
2. This is a processor specification that is unbounded.
3. This is an IMCH specification. The maximum is presently not specified. Intel® 3100 Chipset should not be dependent on this specific value.

The processor goes from a C0 to a C2 state because all of the threads in the processor are idle. The decision to go to the C2 state is made by software.

The following timings are shown in Figure 82.

1. The processor reads the Intel® 3100 Chipset's Level 2 register. Note: Future processors will instead generate a CPU-C2-RDY cycle, which gets forwarded by the IMCH on NSI as a CPU-C2-RDY message. This is not supported by Intel® 3100 Chipset.
2. t1 prior to asserting STPCLK#, Intel® 3100 Chipset will latch the processor interface signals, except SMI# activation due to a synchronous SMI event.
3. t2a later, in response to observing STPCLK# active, the processor(s) performs one or more Stop-Grant cycles on the front-side bus.
4. t2b later, IMCH forwards the last Stop-Grant cycle to Intel® 3100 Chipset via NSI. This will be called "REQ-C2."
5. The processor is now in C2 state. There are some additional steps below that must complete between the IICH and IMCH to keep them in synchronization.
6. t3a after receiving the Stop-Grant from the IMCH, the IICH sends a Go-C2 message to the IMCH
7. t3b after receiving the Go-C2 message, the IMCH sends an Ack-C2 message. At this point, the IMCH is permitted to send the REQ-C0 (Break-Ind) message.

22.7.3.3 C2→C0 Break Sequence

Intel® 3100 Chipset returns the processor to a C0 state in order to execute code. This is due to a break event. See Table 675 for the various break event causes.

The following timings are shown in Figure 82.

1. A Break event is detected by Intel® 3100 Chipset
2. t4a later, Intel® 3100 Chipset sends GO_C0 message to the IMCH.
3. t4b after receiving the Go-C0 message, then IMCH sends an Ack-C0 to the IICH. At this point, the IMCH is not permitted to send the REQ-C0 (Break Ind) Message.
4. t5 after receiving the Ack-C0 message, Intel® 3100 Chipset deasserts STPCLK# to the processor (this enables processor instruction stream)
5. The processor is now back in a C0 state
6. t6 after deasserting STPCLK#, Intel® 3100 Chipset unlatches the processor interface signals, except SMI#, which was not latched for synchronous SMI events.

22.8 Sleep States

22.8.1 Sleep State Overview

Intel® 3100 Chipset directly supports different sleep states (S3, S5), which are entered by setting the SLP_EN bit, or due to a Power Button press. The entry to the Sleep states are based on several assumptions:

- Entry to a Cx state is mutually exclusive with entry to a Sleep state. This is because the processor can only perform one register access at a time. A request to Sleep always has higher priority than throttling.
- Prior to setting the SLP_EN bit, the software turns off processor-controlled throttling. Thermal throttling cannot be disabled, but setting the SLP_EN bit disables thermal throttling (since S3, S5 sleep states have higher priority).
- The G3 state cannot be entered via any software mechanism. The G3 state indicates a complete loss of power.

Table 677 shows the differences in the sleeping states with regard to the listed of Intel® 3100 Chipset output signals:

22.8.2 Initiating Sleep States

Table 677. Sleep State Output Conditions

State	STPCLK#	CPUSLP#	SLP_S3#	SLP_S4#	SLP_S5#
S3	Active	Plane off	Active	Inactive	Inactive
S5	Active	Plane off	Active	Active	Active

Entry to Sleep states (S3, S5) are initiated by any of the following methods:

1. Masking interrupts, turning off all bus master enable bits, setting the desired type in the SLP_TYP field and setting the SLP_EN bit. The hardware will then put the system into the corresponding Sleep state.
2. Pressing the PWRBTN# signal for more than four seconds to cause a Power Button Override event. In this case the transition to the S5 state will be less graceful, since there will be no dependencies on observing Stop-Grant cycles from the processor or on clocks other than the RTC clock.

Other Assumptions:

- Entry to a Cx state is mutually exclusive with entry to a Sleep state. This is because the processor can only perform one register access at a time. A request to Sleep always has higher priority than throttling.

- Setting the SLP_EN bit will disable all throttling (since S3, S5 sleep states have higher priority).
- The G3 state cannot be entered via any software mechanism. The G3 state indicates a complete loss of power.
- Before entering sleep state, an ACPI OS will mask all interrupts and will turn off all bus master enable bits. For non-ACPI systems, the BIOS will mask interrupts and turn off all bus master enable bits. Interrupts might not be masked at the I/O subsystem. Some Operating Systems have been observed to only mask interrupts inside the processor.

Table 678. Sleep Types

Sleep Type	Comment
S3	Intel® 3100 Chipset asserts SLP_S3#. The SLP_S3# signal controls the power to non-critical circuits. Power is only retained to devices needed to wake from this sleeping state, as well as to the memory.
S5	The SLP_S5# signal shuts off the power to the memory subsystem. Only devices needed to wake from this state must be powered. Intel® 3100 Chipset asserts SLP_S3#, SLP_S4# and SLP_S5#.

22.8.3 Exiting Sleep States

Sleep states (S3, S5) are exited based on Wake events. The Wake events will force the system to a full on state (S0), although some non-critical subsystems might still be shut and have to be brought back manually. For example, the hard disk may be shut during a sleep state, and have to be enabled via an I/O pin before it can be used.

Upon exit from Intel® 3100 Chipset-controlled Sleep states, the WAK_STS bit will be set. To enable Wake Events, the possible causes of wake events (and their restrictions) are shown in Table 679.

Table 679. Causes of Wake Events (Sheet 1 of 2)

Cause	States can wake from:	How Enabled
RTC Alarm	S3, S5	Set RTC_EN bit in PM1_EN Register
Power Button	S3, S5	Always enabled as Wake event
GPI[0:15]	S3, S5	GPE0_EN register (after having gone to S5 via SLP_EN, but not after a power failure.) GPIs that are in the core well are not capable of waking the system from sleep states where the core well is not powered.
Classic USB	S3, S5	Set USB1_EN, USB2_EN bits in GPE0_EN Register
RI	S3, S5	Set RI_EN bit in GPE0_EN Register
Primary PME#	S3, S5(Note 2)	PME_B0_EN bit in GPE0_EN register
Secondary PME# (pin)	S3, S5(Note 2)	PME_EN bit in GPE0_EN register.

Notes:

1. If in the S5 state due to a powerbutton override or THRMTRIP#, the only wake events are Power Button, Wake SMBus Slave Message (01h), and Hard Reset SMBus Slave Messages (03h, 04h).
2. PME#, RTC, GPI[0:n], and RI# will be wake events from S5 only if it was entered via software setting the SLP_EN and SLP_TYP bits, or if there is a power failure.
3.

Table 679. Causes of Wake Events (Sheet 2 of 2)

Cause	States can wake from:	How Enabled
SMBus ALERT# Signal	S3, S5	Always Enabled as a Wake Event
SMBus Slave Message	S3, S5, including S5- Power Button Override	Three SMBus commands always enabled as Wake events. These commands (see Note 1. below) can wake from S5 due to Power Button.
SMBus Host Notify message received	S3, S5	HOST_NOTIFY_WKEN bit SMBus Slave Command register. Reported in the SMB_WAK_STS bit in GPE0_STS register.

Notes:

1. If in the S5 state due to a powerbutton override or THRMTRIP#, the only wake events are Power Button, Wake SMBus Slave Message (01h), and Hard Reset SMBus Slave Messages (03h, 04h).
2. PME#, RTC, GPI[0:n], and RI# will be wake events from S5 only if it was entered via software setting the SLP_EN and SLP_TYP bits, or if there is a power failure.
3.

Table 680 summarizes the use of GPIs as wake events.

Table 680. GPI Wake Events

GPI	Power Well	Wake From	Notes
GPI[8]	Resume	S3, S5	ACPI Compliant

The latency to exit the various sleep states varies greatly and is heavily dependent on power supply design, so much so that the exit latencies due to Intel® 3100 Chipset are insignificant.

3.

22.8.4 Sx-G3-Sx, Handling Power Failures

In systems, power failures can occur if the AC power is cut (a real power failure) or if the system is unplugged. In either case, PWROK and RSMRST# are assumed to go low.

Depending on when the power failure occurs and how the system is designed, different transitions could occur due to a power failure.The AFTER_G3 bit provides the ability to program whether or not the system should boot once power returns after a power loss event. If the policy is to not boot, the system remains in an S5 state. There are only three possible events that will wake the system after a power failure.

1. **PWRBTN#:** PWRBTN# is always enabled as a wake event. When RSMRST# is low (G3 state), the PWRBTN_STS bit is reset. When Intel® 3100 Chipset exits G3 after power returns (RSMRST# goes high), the PWRBTN# signal is already high (because V_{CC}-standby goes high before RSMRST# goes high) and the PWRBTN_STS bit is 0.
2. **RI#:** RI# does not have an internal pull-up. Therefore, if this signal is enabled as a wake event, it is important to keep this signal powered during the power loss event. If this signal goes low (active), when power returns the RI_STS bit is set and the system interprets that as a wake event.
3. **RTC Alarm:** The RTC_EN bit is in the RTC well and is preserved after a power loss. Like PWRBTN_STS the RTC_STS bit is cleared when RSMRST# goes low.

Intel® 3100 Chipset monitors both PWROK and RSMRST# to detect for power failures. If PWROK goes low, the PWROK_FLR bit is set. If RSMRST# goes low, PWR_FLR is set. Software can clear PWR_FLR by writing a 1 to that bit

Although PME_EN is in the RTC well, this signal cannot wake the system after a power loss. PME_EN is cleared by RTEST#, and PME_STS is cleared by RSMRST#.

Table 681. Transitions Due To Power Failure

State at Power Failure	AFTERG3_EN bit	Transition When Power Returns
S0, S3	1 0	S5 S0
S5	1 0	S5 S0

The power failure bit (PWR_FLR) is set after any power failure. Software can clear it by writing a 1 to that bit. Refer to Table 651 for information on PWR_FLR.

The CPUPWR_FLR bit separately reports power failures that result in VRMPWRGD going inactive. Refer to Table 652 for information on CPUPWR_FLR.

22.9 Processor Thermal Management

Intel® 3100 Chipset has several mechanisms to assist with managing thermal problems in the system.

22.9.1 THRM# Signal for SMI# or SCI

The THRM# signal is used as a status input from a thermal sensor. The sensor could be inside the processor or in a separate component near the processor. Intel® 3100 Chipset follows these behaviors with regard to the THRM# signal:

1. Based on the THRM# signal going active, Intel® 3100 Chipset generates an SMI# or SCI (depending on SCI_EN).
2. If the THRM_POL bit is set low, when THRM# goes low, the THRM_STS bit will be set. This is an indicator that the thermal threshold has been exceeded. If the THRM_EN bit is set, then when THRM_STS goes active, either an SMI# or SCI# will be generated (depending on the SCI_EN bit being set). The power management software (BIOS or ACPI) can then take measures to start reducing the temperature. Examples include shutting unneeded subsystems, or halting the processor.
3. By setting the THRM_POL bit to high, another SMI# or SCI# can optionally be generated when the THRM# signal goes back high. This allows the software (BIOS or ACPI) to turn off the cooling methods.

Note: THRM# assertion does not cause a TCO event message in S3. The level of the signal is not reported in the heartbeat message.

22.9.2 Processor Initiated Passive Cooling

This is a method to cool the system by throttling the processor. The mode is initiated by software setting the THTL_EN or THTL_DTY bits.

Behavioral Description:

1. Software sets the THTL_DTY bits to select throttle ratio and the THTL_EN bit to enable the throttling.
2. Throttling results in STOPCLK# active for a minimum time of 12.5% and a maximum of 87.5%. The period is 1024 PCI clocks. Thus, the STOPCLK# signal can be active for as little as 128 PCI clocks or as much as 896 PCI clocks. The actual slowdown (and cooling) of the processor will depend on the instruction stream, because the processor is allowed to finish the current instruction. Furthermore,

Intel® 3100 Chipset waits for the STOP-GRANT cycle before starting the count of the time the STOPCLK# signal is active.

3. Intel® 3100 Chipset will perform the Go-C2/Ack-C2 and Go-C0/Ack-C0 messages for throttling, just as if it were making transitions to/from a C2 state.

22.9.3 Force THRM# Throttle Software Bit

The FORCE_THTL bit allows the BIOS to force passive cooling independent of the ACPI software (which uses the THTL_EN and THTL_DTY bits). It has the following behavior:

1. If this bit is set, Intel® 3100 Chipset will start throttling using the ratio in the THRM_DTY field.
2. If this bit is turned off, (cleared)Intel® 3100 Chipset will stop throttling, unless the THTL_EN bit is set (indicating that ACPI software is attempting throttling).

If both the THTL_EN and FORCE_THTL bits are both set, then the IICH must use the duty cycle defined by the THRM_DTY field, not the THTL_DTY field. (i.e., THRM_DTY has higher priority).

22.9.4 Active Cooling

Active cooling involves fans. The GPIO signals from Intel® 3100 Chipset can be used to turn on/off a fan.

22.10 Event Input Signals, Messages and Their Usage

Intel® 3100 Chipset has various input signals that trigger specific events. This section describes those signals and how they should be used.

22.10.1 PWRBTN# – Power Button

Intel® 3100 Chipset PWRBTN# signal operates as a “Fixed Power Button” as described in the *ACPI Specification*. PWRBTN# signal has a 16 ms debounce on the input. The state transition descriptions are included in Table 682. The transitions start as soon as the PWRBTN# is pressed (but after the debounce logic), and does not depend on when the Power Button is released. A power button override will force a transition to S5, even if PWROK is not active.

Table 682. Transitions Due to Power Button

Present State	Event	Transition/Action	Comment
S0/Cx	PWRBTN# goes low	SMI# or SCI generated (depending on SCI_EN)	Software will typically initiate a Sleep state.
S3, S5	PWRBTN# goes low	Wake Event. Transitions to S0 state.	Standard wakeup ***Note:*** Could be impacted by SLP_S4# minimum assertion.
G3	PWRBTN# pressed	None	No effect since no power. Not latched nor detected.
S0,S3	PWRBTN# held low for at least 4 consecutive seconds	Unconditional transition to S5 state.	No dependence on processor (such as Stop-Grant cycles) or any other subsystem.

22.10.1.1 Power Button Override Function

If PWRBTN# is observed active for at least 4 consecutive seconds, then the state machine must unconditionally transition to the G2/S5 state, regardless of present state (S0,S3) even if PWROK is not active. In this case, the transition to the G2/S5 state must not depend on any particular response from the processor (such as a Stop-Grant cycle), nor any similar dependency from any other subsystem.

Note: The 4-second PWRBTN# assertion must only be used if a system lock-up has occurred. The 4-second timer starts counting when Intel® 3100 Chipset is in a S0 state. If the PWRBTN# signal is asserted and held active when the system is in a suspend state (S3,S5), the assertion causes a wake event. Once the system has resumed to the S0 state, the 4-second timer starts.

The PWRBTN# status is readable to check if the button is currently being pressed or has been released. The status is taken after the debounce, and is readable via the PWRBTN_LVL bit.

Note: During the time that the SLP_S4# signal is stretched for the minimum assertion width (if enabled by D31:F0:A4h bit 3), the Power Button is not a wake event. As a result, it is conceivable that the user will press and continue to hold the Power Button waiting for the system to awake. Since a 4-second press of the Power Button is already defined as an Unconditional Power down, the power button timer will be forced to inactive while the power-cycle timer is in progress. Once the power-cycle timer has expired, the Power Button awakes the system. Once the minimum SLP_S4# power cycle expires, the Power Button must be pressed for another 4 to 5 seconds to create the Override condition to S5.

22.10.1.2 Sleep Button

The *Advanced Configuration and Power Interface (ACPI) Specification, Rev. 2.0b* defines an optional Sleep button. It differs from the power button in that it only is a request to go from S0 to S3 (not S5). Also, in an S5 state, the Power Button can wake the system, but the Sleep Button cannot.

Although Intel® 3100 Chipset does not include a specific signal designated as a Sleep Button, one of the GPIO signals can be used to create a “Control Method” Sleep Button See the *ACPI Specification* for implementation details.

22.10.2 RI# – Ring Indicate Signal

The Ring Indicator can cause a wake event (if enabled) from the S3, S5 states. Table 683 shows when the wake event is generated or ignored in different states. If in the G0/S0/Cx states, Intel® 3100 Chipset generates an interrupt based on RI# active, and the interrupt will be set up as a Break event.

Note: There is no filtering on the RI# signal. Any debounce filtering must be done externally.

Table 683. Transitions Due to RI# Signal

Present State	Event	RI_EN	Event
S0	RI# Active	X	Ignored
S3, S5	RI# Active	0 1	Ignored Wake Event

22.10.3 PME# – PCI Power Management Event

The PME# signal comes from a PCI device to request that the system be restarted. The PME# signal can generate an SMI, SCI, or optionally a Wake event. The event occurs when the PME# signal goes from high to low. No event is caused when it goes from low to high.

22.10.4 SYS_RESET# Button

When the SYS_RESET# button is detected as active after the debounce logic (16 ms debounce on the input, same as PWRBTN#), Intel® 3100 Chipset will attempt to perform a "graceful" reset, by waiting up to 25 ms, +/- 2ms for SMBus to go idle. If SMBus is idle when the button is detected active, the reset will occur immediately, otherwise the counter will start. If at any point during the count SMBus goes idle, the counter will be reset and the full system reset will occur. If, however, the counter expires and SM Bus is still active, a full system reset will be forced upon the system even though SMBus activity is still occurring.

Once the reset is asserted, it will remain asserted for approximately 1 ms, regardless of whether the SYS_RESET# input remains asserted or not. It cannot occur again until SYS_RESET# has been detected inactive after the debounce logic, and the system is back to a full S0 state with PLTRST# inactive.

Note: If bit 3 of the CF9h I/O register is set then SYS_RESET# will result in a full power cycle reset.

22.10.5 Processor Thermal Trip

If THRMTRIP# goes active, the processor is indicating an overheat condition, and Intel® 3100 Chipset will immediately transition to an S5 state. However, since the processor has overheated, it will not respond to Intel® 3100 Chipset's STPCLK# pin with a stop grant special cycle. Therefore, Intel® 3100 Chipset will not wait for one. Immediately upon seeing THRMTRIP# low, Intel® 3100 Chipset will initiate a transition to the S5 state, drive signals SLP_S3#, SLP_S4#, SLP_S5# low, and set the CTS bit. The transition will generally look like a power button override.

It is extremely important that when a THRMTRIP# event occurs, Intel® 3100 Chipset power down immediately without following the normal S0 -> S5 path. This can fire in parallel, but Intel® 3100 Chipset must immediately enter a power down state. It will do this by driving signals SLP_S3#, SLP_S4#, and SLP_S5# within 3 PCICLKs after sampling THRMTRIP# active.

If the processor is running extremely hot and is heating up, it is possible (although very unlikely) that components around it, such as the IICH, are no longer executing cycles properly. Therefore, if THRMTRIP# goes active, and IICH is relying on state machine logic to perform the power down, the state machine may not be working, and the system will not power down.

Intel® 3100 Chipset will follow this flow for THRMTRIP#.

1. At boot (PLTRST# low), THRMTRIP# ignored.
2. After power-up (PLTRST# high), if THRMTRIP# sampled active, SLP_S3#, SLP_S4#, and SLP_S5# fire, and normal sequence of sleep machine starts.
3. Until sleep machine enters the S5 state, SLP_S3#, SLP_S4#, and SLP_S5# stay active, even if THRMTRIP# is now inactive. This is the equivalent of "latching" the thermal trip event.
4. When S5 state reached, go to step #1, otherwise stay here. If Intel® 3100 Chipset never gets to S5, Intel® 3100 Chipset does not reboot until power is cycled.

During boot, THRMTRIP# is ignored until SLP_S3#, PWROK, VRMPWRGD/VGATE, and PLTRST# are all '1'. During entry into a powered-down state (due to S3, S5 entry, power cycle reset, etc.) THRMTRIP# is ignored until either SLP_S3# = 0, or PWROK = 0, or VRMPWRGD/VGATE = 0.

Note: A processor thermal trip event will

1. Set the AFTERG3_EN bit
2. Clear the PWRBTN_STS bit
3. Clear all the GPE0_EN register bits
4. Clear the SMB_WAK_STS bit only if SMB_WAK_STS was set due to SMBus slave receiving message and not set due to SMBAlert.

Note: The THRMTRIP# pin must be glitch free.

22.10.6 SATA SCI

The SATA logic can cause an SCI, but not an SMI or wake event. When the SATA logic causes an SCI, the SATA_SCI_STS bit will be set. The SCI handler enables the SCI and clears the SCI via bits in the SATA unit.

22.10.7 PCI Express WAKE# Signal and PME Event Message

PCI Express ports can wake the platform from any sleeping state (S3 or S5) using the PCI-EXP-WAKE# pin. The WAKE# signal is treated as a wake event, but does not cause any bits to go active in the GPE_STS register.

PCI Express ports and the IMCH (via NSI) have the ability to cause PME using messages. When a PME message is receive, Intel® 3100 Chipset will set the PCI-EXP-STS bit.

22.10.8 PCI Express Hot Plug

PCI Express has a hot plug mechanism. See Section 30.3 for further details on the bits associated with each PCI Express port.

The Hot Plug logic is capable of generating an SCI (via the GPE1 register). It is also capable of generating an SMI. However, it is not capable of generating a wake event.

22.11 Alternate (ALT) Access Mode

Before entering a low power state, several registers from powered down parts may need to be saved. In the majority of cases, this is not an issue, as registers have read and write paths. However, several of the ISA (legacy) compatible registers are either read-only or write-only. To get data out of write-only registers, and to restore data into read-only registers, the IICH implements an alternate access mode.

If the ALT access mode is entered and exited after reading the registers of IICH timer (8254), the timer starts counting faster (13.5 ms). The following steps listed below can cause problems:

1. BIOS enters ALT access mode for reading the IICH timer related registers.
2. BIOS exits ALT access mode.
3. BIOS continues through the execution of other needed steps and passes control to the operating system.

After getting control in step #3, if the operating system does not reprogram the system timer again, the timer ticks may be happening faster than expected. For example DOS and its associated software assume that the system timer is running at 54.6 ms and as a result the time-outs in the software may be happening faster than expected.

Operating systems (e.g., Microsoft Windows* 98, Windows* 2000, and Windows NT*) reprogram the system timer and therefore do not encounter this problem.

For some other loss (e.g., Microsoft MS-DOS*) the BIOS must restore the timer back to 54.6 ms before passing control to the operating system. If the BIOS is entering ALT access mode before entering the suspend state it is not necessary to restore the timer contents after the exit from ALT access mode.

22.11.1 Write Only Registers with Read Paths in Alternate Access Mode

The registers described in Table 684 have read paths in alternate access mode. The access number field in the table indicates which register will be returned per access to that port.

Table 684. Write-Only Registers with Read Paths in Alternate Access Mode (Sheet 1 of 2)

Restore Data						Restore Data	
I/O Addr	# of Reads	Access	Data	I/O Addr	# of Reads	Access	Data
00h	2	1	DMA Chan 0 base address low byte	40h	7	1	Timer Counter 0 status, bits [5:0]
		2	DMA Chan 0 base address high byte			2	Timer Counter 0 base count low byte
01h	2	1	DMA Chan 0 base count low byte			3	Timer Counter 0 base count high byte
		2	DMA Chan 0 base count high byte			4	Timer Counter 1 base count low byte
02h	2	1	DMA Chan 1 base address low byte			5	Timer Counter 1 base count high byte
		2	DMA Chan 1 base address high byte			6	Timer Counter 2 base count low byte
03h	2	1	DMA Chan 1 base count low byte			7	Timer Counter 2 base count high byte
		2	DMA Chan 1 base count high byte	41h	1		Timer Counter 1 status, bits [5:0]
04h	2	1	DMA Chan 2 base address low byte	42h	1		Timer Counter 2 status, bits [5:0]
		2	DMA Chan 2 base address high byte	70h	1		Bit 7 = NMI Enable, Bits [6:0] = RTC Address
05h	2	1	DMA Chan 2 base count low byte	C4h	2	1	DMA Chan 5 base address low byte
		2	DMA Chan 2 base count high byte			2	DMA Chan 5 base address high byte
06h	2	1	DMA Chan 3 base address low byte	C6h	2	1	DMA Chan 5 base count low byte
		2	DMA Chan 3 base address high byte			2	DMA Chan 5 base count high byte

Notes:
1. The OCW1 register must be read before entering Alternate Access Mode.
2. Bits 5, 3, 1, and 0 return 0.
3. The additional write-only registers are described in their respective sections.

Table 684. Write-Only Registers with Read Paths in Alternate Access Mode (Sheet 2 of 2)

Restore Data						Restore Data	
I/O Addr	# of Reads	Access	Data	I/O Addr	# of Reads	Access	Data
07h	2	1	DMA Chan 3 base count low byte	C8h	2	1	DMA Chan 6 base address low byte
		2	DMA Chan 3 base count high byte			2	DMA Chan 6 base address high byte
08h	6	1	DMA Chan 0-3 Command[2]	CAh	2	1	DMA Chan 6 base count low byte
		2	DMA Chan 0-3 Request			2	DMA Chan 6 base count high byte
		3	DMA Chan 0 Mode: Bits(1:0) = “00”	CCh	2	1	DMA Chan 7 base address low byte
		4	DMA Chan 1 Mode: Bits(1:0) = “01”			2	DMA Chan 7 base address high byte
		5	DMA Chan 2 Mode: Bits(1:0) = “10”	CEh	2	1	DMA Chan 7 base count low byte
		6	DMA Chan 3 Mode: Bits(1:0) = “11”.			2	DMA Chan 7 base count high byte
20h	12	1	PIC ICW2 of Master controller	D0h	6	1	DMA Chan 4-7 Command[2]
		2	PIC ICW3 of Master controller			2	DMA Chan 4-7 Request
		3	PIC ICW4 of Master controller			3	DMA Chan 4 Mode: Bits(1:0) = “00”
		4	PIC OCW1 of Master controller[1]			4	DMA Chan 5 Mode: Bits(1:0) = “01”
		5	PIC OCW2 of Master controller			5	DMA Chan 6 Mode: Bits(1:0) = “10”
		6	PIC OCW3 of Master controller			6	DMA Chan 7 Mode: Bits(1:0) = “11”.
		7	PIC ICW2 of Slave controller				
		8	PIC ICW3 of Slave controller				
		9	PIC ICW4 of Slave controller				
		10	PIC OCW1 of Slave controller[1]				
		11	PIC OCW2 of Slave controller				
		12	PIC OCW3 of Slave controller				

Notes:
1. The OCW1 register must be read before entering Alternate Access Mode.
2. Bits 5, 3, 1, and 0 return 0.
3. The additional write-only registers are described in their respective sections.

22.11.2 PIC Reserved Bits

Many bits within the PIC are reserved, and must have certain values written in order for the PIC to operate properly. Therefore, there is no need to return these values in alternate (ALT) access mode. When reading PIC registers from 20h and A0h, the reserved bits shall return the values listed in Table 685.

Table 685. PIC Reserved Bits Return Values

PIC Reserved Bits	Value Returned
ICW2(2:0)	000
ICW4(7:5)	000
ICW4(3:2)	00
ICW4(0)	0
OCW2(4:3)	00
OCW3(7)	0
OCW3(5)	Reflects bit 6
OCW3(4:3)	01

22.11.3 Read-Only Registers with Write Paths in ALT Access Mode

The registers described in Table 686 have write paths to them in ALT access mode. Software will restore these values after returning from a powered down state. These registers must be handled special by software. When in normal mode, writing to the base address/count register also writes to the current address/count register. Therefore, the base address/count must be written first, then the part is put into alternate access mode and the current address/count register is written.

Only bits 3:0 of the DMA Status Registers listed below are writable.

Table 686. Register Write Accesses in Alternate Access Mode

I/O Address	Register Write Value
08h	DMA Status Register for channels 0-3.
D0h	DMA Status Register for channels 4-7.

22.12 System Power Supplies, Planes, and Signals

22.12.1 Power Plane Control with SLP_S3#, SLP_S4# and SLP_S5#

The usage of SLP_S3# and SLP_S4# depend on whether the platform is configured for S3-Hot or S3-Cold.

- S3-Hot — The SLP_S3# signal is used to cut power only to the processor and to stop the system clocks.
- S3-Cold — The SLP_S3# output signal can be used to cut power to the system core supply, since it will only go active for the STR state (typically mapped to ACPI S3). Power must be maintained to system memory, Intel® 3100 Chipset Resume Well, and to any other circuits that need to generate Wake signals from the STR state.

Cutting power to the core may be done via the power supply, or by external FETs to the motherboard.

The SLP_S4# and SLP_S5# output signal can be used to cut power to the system core supply, as well as power to the system memory, since the context of the system is saved on the disk. Cutting power to the memory may be done via the power supply, or by external FETs to the motherboard. In systems set up for S3-Hot, the SLP_S4# is also used to kill power to the subsystems that are powered during the S3-Hot state.

The SLP_S4# output signal is used to remove power to additional subsystems that are powered during SLP_S3#.

SLP_S5# output signal can be used to cut power to the system core supply, as well as power to the system memory, since the context of the system is saved on the disk. Cutting power to the memory may be done via the power supply, or by external FETs to the motherboard.

22.12.2 SLP_S4# and Suspend-To-RAM Sequencing

The system memory suspend voltage regulator is controlled by Glue logic. The SLP_S4# signal must be used to remove power to system memory rather than the SLP_S5# signal. The SLP_S4# logic in Intel® 3100 Chipset provides a mechanism to fully cycle the power to the DRAM and/or detect if the power is not cycled for a minimum time.

Note: To utilize the hardware-enforced minimum DRAM power-down feature that is enabled by the SLP_S4# Assertion Stretch Enable bit (Section 22.5.1.3, "Offset A4h: GEN_PMCON_3 – General PM Configuration 3 Register" bit 3), the DRAM power must be controlled by the SLP_S4# signal.

22.12.3 PWROK Signal

The PWROK input must go active based on the core supply voltages becoming valid. PWROK must not go high until at least 99 ms after Vcc3_3 and Vcc1_5 have reached their nominal values.This is required to meet the 100 ms delay from valid power to PLTRST# deassertion in the *PCI Specification, Rev. 2.3*.

1. Traditional designs have an active-low reset button electrically ANDed with the PWROK signal from the power supply and the processors voltage regulator module. If this is done with Intel® 3100 Chipset, the PWROK_FLR bit will be set. Intel® 3100 Chipset treats this internally as if the RSMRST# signal had gone active. However, it is not treated as a full power failure. If PWROK goes inactive and then active (but RSMRST# stays high), then Intel® 3100 Chipset will reboot (regardless of the state of the AFTERG3 bit). If the RSMRST# signal also goes low before PWROK goes high, then this is a full power failure, and the reboot policy is controlled by the AFTERG3 bit.
2. SYSRESET# is recommended for implementing the system reset button. This saves the external logic that is needed when the PWROK input is used. Additionally it allows for better handling of the SM-Bus and processor resets, and avoids improperly reporting or power failures.
3. PWROK and RSMRST# are sampled using the RTC clock. Therefore, low times that are less than one RTC clock period may not be detected by Intel® 3100 Chipset.
4. In the case of true PWROK failure, PWROK will go low first before the VRMPWRGD.
5. If the PWROK input is used to implement the system reset button, Intel® 3100 Chipset does not provide any mechanism to limit the amount of time that the processor is held in reset. The platform must externally guarantee that maximum reset assertion specs are met.

22.12.4 CPUPWRGD Signal

This signal is connected to the processor and is derived from two inputs: VRMPWRGD signal (from the processor's VRM) AND'd with the PWROK signal that comes from the system power supply.

Warning: Intel® 3100 Chipset has no specific mechanism to prevent the CPU RESET signal from being held active too long. Either the IMCH or the platform design must be designed to meet this requirement.

22.12.5 Controlling Leakage and Power Consumption During Low-Power States

To control leakage in the system, various signals will tri-state or go low during some low-power states.

General principles (these are board-level guidelines and are NOT Intel® 3100 Chipset behavioral rules):

- All signals going to powered down planes (either internal or external) must be either tri-states or driven low.
- Signals with pull-up resistors must not be low during low-power states. This is to avoid the power consumed in the pull-up resistor.
- Buses must be halted (and held) in a known state to avoid a floating input (perhaps to some other device). Floating inputs can cause extra power consumption.

Based on the above principles, the following measures are taken:

- During S3 (STR), all signals attached to powered down planes will be tri-stated or driven low.

22.12.6 VRMPWROK

The VRMPWROK signal is generated by the processor's VRM. It indicates that the voltage outputs from the VRM are on and within spec. VRMPRWOK may go active before or after the PWROK from the main power supply. Intel® 3100 Chipset has no dependency on the order in which these two signals go active or inactive

22.13 Legacy Power Management Theory of Operation

22.13.1 Overview

Instead of relying on ACPI software, legacy power management uses BIOS and various hardware mechanisms. The scheme relies on the concept of detecting when individual subsystems are idle, detecting when the whole system is idle, and detecting accesses are attempted to idle subsystems.

However, the OS is assumed to at least be APM enabled. Without APM calls, there is no quick way to know when the system is idle between keystrokes.Intel® 3100 Chipset does not support burst modes.

22.13.2 APM Power Management

Intel® 3100 Chipset has a timer, when enabled by the 1MIN_EN bit in the SMI Control and Enable register, generates a periodic SMI# once per minute.In Intel® 3100 Chipset, there is also an option to have it generate the SMI# once per 32, 16, or 8

seconds.The SMI handler can check for system activity by reading the DEVTRAP_STS register. If none of the system bits are set, the SMI handler can increment a software counter.

If there is activity, the various bits in the DEVTRAP_STS register will be set. Software clears the bits by writing a 1 to the bit position.

The DEVTRAP_STS register allows for monitoring of various internal devices or Super I/O devices (SP, PP, FDC) on LPC or PCI, keyboard controller accesses, or audio functions on PCI or LPC. Other PCI activity can be monitored by checking the PCI Interrupts.

23.0 System Management

23.1 Overview

The Intel® 3100 Chipset provides various functions to make a system easier to manage and lower the Total Cost of Ownership (TCO) of the system. Features and functions can be augmented via external A/D converters and GPIO, as well as an external microcontroller.

The following features and functions are supported by the Intel® 3100 Chipset:

- **First timer to generate SMI# after programmable time**
  - First timeout causes SMI#; allows for SMM-based recovery from operating system lockup
  - Operating system-based software agent accesses the Intel® 3100 Chipset to periodically reload timer
- **Ability for SMM handler to generate "TCO" interrupt to operating system**
  - Allows for operating system-based code augmentation
- **Ability for operating system to generate SMI#**
  - Call-back from operating system to TCO code in SMM handler
- **Second hard coded timeout to generate reboot**
  - Used only after first timeout occurs
  - Second timeout allows for automatic system reset and reboot if hardware error detected. Various system states are preserved via this special reset to allow for possible error detection and correction.
  - Reset associated with reboot may attempt to preserve some registers for diagnostic purposes
  - SMI# handler must reload the main timer within 2.4 s to prevent the second timer from causing a reboot (timeout during SMI is assumed as broken CPU or stuck hardware)
  - Option to prevent reset if second timeout occurs
- Processor present detection
  - Detects if processor fails to fetch the first instruction after reset
  - If CPU failure detected, option to pulse a GPIO or send SMBus message. The SMBus message can be used to indicate to an External LAN controller to send a distress message. The GPIO can control an LED with optional blink.
- **Ability to handle various errors (such as ECC errors) indicated by the** IMCH
  - Can generate SMI# or TCO interrupt
- **Intruder detect input**
  - Can generate TCO interrupt or SMI# when the system cover is removed
- **Ability for TCO messages to coexist with standard SMBus devices**

- **Detection of bad FWH programming**
  - Detects if data on first read is FFh (indicates unprogrammed Firmware Hub)

23.2 TCO Register Map

The TCO logic is accessed via registers mapped to the PCI configuration space (Device 31, Function 0) and the system I/O space. For TCO PCI Configuration registers, see LPC Device 31, Function 0: PCI Configuration registers.

23.2.1 TCO PCI Configuration Registers

See Chapter 16.0, "Device 31, Function 0: LPC Interface."

Allows setting of the Base Address for the I/O space and routing of the TCO interrupt.

23.2.2 TCO I/O-Mapped Registers

The TCO I/O registers reside in a 32-byte range that starts 96 bytes above the power management (ACPI) I/O space (Section 16.2.2.1, "Offset 40 - 43h: ABASE – ACPI Base Address Register"). Thus TCOBASE =ACPIBASE + 60h in the PCI configuration space. Table 688 shows the mapping of the registers within that 32-byte range. Each register is described in the following sections.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 687. TCO I/O Registers Summary Table

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	01h	TRLD	TCO Timer Reload and Current Value Register	0000h	RW
02h	02h	TDI	TCO Data In Register (from the operating system to the SMI Handler)	00h	RW
03h	03h	TDO	TCO Data Out Register (from SMI Handler to the operating system)	00h	RW
04h	05h	TSTS1	TCO 1 Status Register	0000h	RW
06h	07h	TSTS2	TCO 2 Status Register	0000h	RW
08h	09h	TCTL1	TCO 1 Control Register	0000h	RW
0Ah	0Bh	TCTL2	TCO 2 Control Register	0008h	RW
0Ch	0Dh	TMSG	TCO Message 1 and 2 Register	00h	RW
0Eh	0Eh	TWDS	TCO Watchdog Status Register	00h	RW
0Fh	0Fh	RSV	Reserved	00h	RO
10h	10h	LE	Legacy Elimination Register	11h	RW
12h	13h	TTMR	TCO Timer Initial Value Register	0004h	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

23.2.2.1 Offset 00 - 01h: TRLD – TCO Timer Reload and Current Value Register

Table 688. Offset 00 - 01h: TRLD – TCO Timer Reload and Current Value Register

I/O Offset: *TCOBASE + (00 - 01h)*			***Size:*** *16 bits*	
Default Value: *0000h*			***Power Well:*** *Core*	
Bits	**Name**	**Description**	**Reset Value**	**Access**
15:10	Reserved	**Reserved**.	00h	
09:00	TRLD	**TCO Timer Value:** Reading this register returns the current count of the TCO timer. Writing any value to this register reloads the timer to prevent the timeout.	000h	RW

23.2.2.2 Offset 02h: TDI – TCO Data In Register

Table 689. Offset 02h: TDI – TCO Data In Register

I/O Offset: *TCOBASE +02h*			***Size:*** *8 bit*	
Default Value: *00h*			***Power Well:*** *Core*	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:00	TDI	This data register field is used for passing commands from the operating system to the SMI handler. Writes to this register cause an SMI and set the OS_TCO_SMI bit in the TCO_STS register (D31, F0, 04h).	00h	RW

23.2.2.3 Offset 03h: TDO – TCO Data Out Register

Table 690. Offset 03h: TDO – TCO Data Out Register

I/O Offset: *TCOBASE +03h*			***Size:*** *8 bit*	
Default Value: *00h*			***Power Well:*** *Core*	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:00	TDO	This data register field is used for passing commands from the SMI handler to the operating system. Writes to this register sets the TCO_INT_STS bit in the TCO_STS register. It also causes an interrupt, as selected by the TCO_IRQ_SEL bits.	00h	RW

23.2.2.4 Offset 04 - 05h: TSTS1 – TCO 1 Status Register

Unless otherwise indicated, these bits are “sticky” and are cleared by writing a one to the corresponding bit position.

Table 691. Offset 04 - 05h: TSTS1 – TCO 1 Status Register (Sheet 1 of 2)

I/O Offset: *TCOBASE + (04 - 05h)* **Size:** *16 bit*
Default Value: *0000h* **Power Well:** *Core*

Bits	Name	Description	Reset Value	Access
15:13	Reserved	Reserved	000	
12	MCHSERR_STS	0 = Software clears this bit by writing a 1 to it. 1 = The IMCH sent a NSI special cycle message via NSI indicating that it wants to cause an SERR#. The software must read the IMCH to determine the reason for the SERR#.	0	RWC
11	Reserved	Reserved.	0	
10	MCHSMI_STS	0 = Software clears this bit by writing a 1 to it. 1 = IMCH sends a NSI special cycle message via NSI indicating that it wants to cause an SMI. The software must read the IMCH to determine the reason for the SMI.	0	RWC
09	MCHSCI_STS	0 = Software clears this bit by writing a 1 to it. 1 = IMCH sends a NSI special cycle message via NSI indicating that it wants to cause an SCI. The software must read the IMCH to determine the reason for the SCI.	0	RWC
08	BIOSWR_STS	0 = Software clears this bit by writing a 1 to it. 1 = The Intel® 3100 Chipset sets this bit and generates and SMI# to indicate an illegal attempt to write to the BIOS. This occurs when either: a) The BIOSWP bit is changed from 0 to 1 and the BLD bit is also set, or b) any write is attempted to the BIOS and the BIOSWP bit is also set. ***Note:*** On write cycles attempted to the 4 Mbyte lower alias to the BIOS space, the BIOSWR_STS is not set.	0	RWC
07	NEWCENTURY_STS	This bit is in the RTC well. 0 = Cleared by writing a 1 to the bit position or by RTEST# going active. 1 = This bit is set when the Year byte (index offset 09h) rolls over from 1999 to 2000. If the bit is already 1, it remains 1. When this bit is set, an SMI# is generated. However, this is not a wake event (i.e., if the system is in a sleeping state when the NEWCENTURY_STS bit is set, the system does not wake up). ***Note:*** The NEWCENTURY_STS is not valid when the RTC battery is first installed (or if the RTC battery does not provide sufficient power when the system is unplugged). Software can determine that the RTC well was not maintained by checking the RTC_PWR_STS bit (D31:F0:A4, bit 2) or by other means (such as doing a checksum on the RTC RAM array). If the RTC power is determined to not have been maintained, the BIOS must set the time to a legal value and then clear the NEWCENTURY_STS bit. The NEWCENTURY_STS bit may take up to 3 RTC clocks for the bit to be cleared when a 1 is written to the bit to clear it. After writing a 1 to the NEWCENTURY_STS bit, software must not exit the SMI handler until verifying that the bit has actually been cleared. This ensures that the SMI is not reentered.	0	RWC
06:04	Reserved	Reserved	0	
03	TIMEOUT	0 = Software clears this bit by writing a 1 to it. 1 = Set to indicate that the SMI was caused by the TCO timer reaching 0. ***Note:*** The SMI handler must clear this bit to prevent an immediate reentry to the SMI handler.	0	RWC

Table 691. Offset 04 - 05h: TSTS1 – TCO 1 Status Register (Sheet 2 of 2)

I/O Offset: *TCOBASE + (04 - 05h)*		**Size:** *16 bit*		
Default Value: *0000h*		**Power Well:** *Core*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
02	TCO_INT_STS	0 = Software clears this bit by writing a 1 to it. 1 = SMI handler caused the interrupt by writing to the TCO_DAT_OUT register (TCOBASE + 03h).	0	RWC
01	OS_TCO_SMI	0 = Software clears this bit by writing a 1 to it. 1 = Software caused an SMI# by writing to the TCO_DAT_IN register (TCOBASE + 02h).	0	RWC
00	NMI2SMI_STS	0 = Cleared by clearing the associated NMI# status bit. 1 = Set when an SMI# occurs because an event occurred that would otherwise have caused an NMI#. ***Note:*** The NMI2SMI_STS bit must not be "sticky bit". It must be a simple OR gate to indicate that one of the NMI sources has caused the SMI. Each of the NMI sources already has its own sticky bit feeding the OR gate. ***Note:*** Writes to this bit have no effect.	0	RWC

23.2.2.5 Offset 06 - 07h: TSTS2 – TCO 2 STS Register

Table 692. Offset 06 - 07h: TSTS2 – TCO 2 STS Register (Sheet 1 of 2)

I/O Offset: *TCOBASE + (06 - 07h)*		**Size:** *16 bit*		
Default Value: *0000h*		**Power Well:** *Resume*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
15:05	Reserved	Reserved	000h	
04	SMLINK_ SLAVE_ SMI_STS	Allows the software to go directly into predetermined sleep state. This avoids race conditions. Software clears this bit by writing a 1 to it. 0 = The bit is reset by RSMRST#, but not due to the PCI Reset associated with exit from S3–S5 states. 1 = The Intel® 3100 Chipset sets this bit to 1 when it receives the SMI message (encoding 08h in the command type) on the SMLink's Slave Interface. This bit is in the resume well. It is reset by RSMRST#–	0	RWC
03	BAD_BIOS	This bit is not intended to be read by the BIOS or software. It is only used for sending the TCO messages to an External LAN Controller. 0 = The first BIOS read is not FFh. This is detected when the initial read returns FFh from the FWH. Reads to this bit always return 0 and writes have no effect. 1 = FFh is detected on the first BIOS read (i.e., the BIOS is bad).	0	RW

Table 692. Offset 06 - 07h: TSTS2 – TCO 2 STS Register (Sheet 2 of 2)

I/O Offset: TCOBASE + (06 - 07h)	**Size:** 16 bit
Default Value: 0000h	**Power Well:** Resume

Bits	Name	Description	Reset Value	Access
02	DOACPU_ STS	0 = Cleared based on RSMRST# or by software writing a 1 to this bit. Software must first clear the SECOND_TO_STS bit before writing a 1 to clear the BOOT_STS bit. 1 = Set to 1 when the SECOND_TO_STS bit goes from 0 to 1 and the processor has not fetched the first instruction. If rebooting due to a SECOND_TO_STS bit set (= 1) and the DOACPU_STS bit is: 0 = The BIOS can conclude that the system rebooted due to some lockup (such as on NSI), but not due to a processor booting issue. 1 = Reboots using the 'safe' multiplier (1111). This allows the system to recover from a processor frequency multiplier that is too high, and allows the BIOS to check the DOACPU_STS bit at boot. If the bit is set and the frequency multiplier is 1111, then the BIOS knows that the processor has been programmed to an illegal multiplier. ***Note:*** Software must clear the SECOND_TO_STS bit first, then the DOACPU_STS bit (use two separate I/O write operations).	0	RWC
01	SECOND_ TO_STS	0 = Software clears this bit by writing a 1 to it or by a RSMRST#. 1 = Sets this bit to 1 to indicate that the TIMEOUT bit had been (or is currently) set and a second timeout occurred before the TCO_RLD register was written. If this bit is set and the NO_REBOOT configuration bit is 0, then reboots the system after the second timeout. The reboot is done by asserting PLTRST#.	0	RWC
00	INTRD_DET	**Intruder Detect.** This bit resides in the RTC well. 0 = Software clears this bit by writing a 1 to this bit or by RTEST#. 1 = Set to indicate that an intrusion was detected. This bit is latched. The INTRUDER# signal must be asserted for a minimum of 1 ms to guarantee that the INTRD_DET bit is set. This bit has a recovery time. After writing a 1 to this bit position (to clear it), the bit may be read back as a 1 for up 65 µs before it is read as a 0. Software must be aware of this recovery time when reading this bit after clearing it. If the INTRUDER# signal is active when the software attempts to clear the INTRD_DET bit, the bit remains one, and the SMI# is generated again immediately. The SMI handler can clear the INTRD_SEL bits (TCOBASE + 0Ah, bits 2:1) to avoid further SMIs. However, if the INTRUDER# signals goes inactive and then active again, there is not further SMIs (because the INTRD_SEL bits would select that no SMI# be generated). If the INTRUDER# signal goes inactive some point after the INTRD_DET bit is written as a 1, then the INTRD_DET signal goes to a 0 when INTRUDER# input signal goes inactive. This is slightly different than a classic sticky bit, since most sticky bits would remain active indefinitely when the signal goes active and would immediately go inactive when a 1 is written to the bit.	0	RWC

23.2.2.6 Offset 08 - 09h: TCTL1 – TCO 1 Control Register

Table 693. Offset 08 - 09h: TCTL1 – TCO 1 Control Register

I/O Offset: *TCOBASE + (08 - 09h)* ***Size:*** *16 bit*

Default Value: *0000h* ***Power Well:*** *Core*

Bits	Name	Description	Reset Value	Access
15:13	Reserved	Reserved	000	
12	TCO_LOCK	0 = A core well reset is required to change this bit from 1 to 0. This bit defaults to 0. 1 = This bit prevents writes from changing the TCO_EN bit (in offset 30h of Power Management I/O space). Once this bit is set to 1, it can not be cleared by software writing a 0 to this bit location.	0	RW
11	TCO_TMR_ HALT	0 = The TCO timer is enabled to count. 1 = The TCO Timer halts. It does not count, and thus cannot reach a value that causes an SMI# or set the SECOND_TO_STS bit. When set, this bit prevents rebooting.	0	RW
10	SEND_NOW	0 = Clears this bit when it has completed sending the message. Warning: Software must not set this bit to 1 again until the Intel® 3100 Chipset has set it back to 0. 1 = Set the SEND_NOW bit and causes the LAN controller to reset, which can have unpredictable side-effects. Unless software protects against these side-effects, software must not set or otherwise use the SEND_NOW bit.	0	RW
09	NMI2SMI_EN	0 = Normal NMI functionality 1 = Setting this bit 1 forces all NMIs to instead cause an SMI#, and is reported in the TCO1_STS register. NMI2SMI_EN bit is set AND the NMI_EN bit is set, the NMI# is routed to cause an SMI#. No NMI is caused. However, if the GBL_SMI_EN bit is not set, then no SMI# is generated, either. If NMI2SMI_EN is set but the NMI_EN bit is not set, then no NMI or SMI# is generated. The following table shows the possible combinations: NMI_EN / GBL_SMI_EN / Description 0b / 0b / No SMI# at all because GBL_SMI_EN = 0 0b / 1b / SMI# is caused due to NMI events 1b / 0b / No SMI# at all because GBL_SMI_EN = 0 1b / 1b / No SMI# due to NMI because NMI_EN = 1	0	RW
08	NMI_NOW	0 = Software clears this bit by writing a 1 to it. The NMI handler is expected to clear this bit. Another NMI is not generated until the bit is cleared. 1 = Writing a 1 to this bit causes an NMI. This allows the BIOS or SMI handler to force an entry to the NMI handler.	0	RWC
07:00	Reserved	Reserved	00h	

23.2.2.7 Offset 0A - 0Bh: TCTL2 – TCO 2 Control Register

Table 694. Offset 0A - 0Bh: TCTL2 – TCO 2 Control Register

I/O Offset: *TCOBASE + (0A - 0Bh)* ***Size:*** *16 bit*

Default Value: *0008h* ***Power Well:*** *Resume*

Bits	Name	Description	Reset Value	Access
15:06	Reserved	Reserved	000h	
05:04	OS_POLICY	Operating system-based software writes to these bits to select the policy that the BIOS uses after the platform resets due the WDT. The following convention is recommended for the BIOS and operating system: 00 Boot normally 01 Shut down 10 Do not load operating system. Hold in preboot state and use LAN to determine next step 11 Reserved ***Note:*** These are scratchpad bits. They must not be reset when the TCO logic resets the platform due to Watchdog Timer.	00	RW
03	GPIO11_ ALERT_ DISABLE	At reset (via RSMRST# asserted) this bit is set and GPI[11] alerts are disabled. 0 = Enable 1 = Disable GPI[11]/SMBALERT# as an alert source for the SMBus slave	1	RW
02:01	INTRD_SEL	This field selects the action to take if the INTRUDER# signal goes active. 00 No interrupt or SMI# 01 Interrupt (as selected by TCO_INT_SEL). 10 SMI# 11 Reserved	00	RW
00	Reserved	Reserved	0	

23.2.2.8 Offset 0C - 0Dh: TMSG – TCO MESSAGE 1 and 2 Registers

Table 695. Offset 0C - 0Dh: TMSG – TCO MESSAGE 1 and 2 Registers

I/O Offset: *TCOBASE +0Ch - Message1*
TCOBASE +0Dh - Message 2 ***Size:*** *8 bit*

Default Value: *00h* ***Power Well:*** *Resume*

Bits	Name	Description	Reset Value	Access
07:00	TMSG	The value written into this register is sent out in the MESSAGE field of the SMBus Event messages. BIOS can write to this register to indicate its boot progress which can be monitored externally.	00h	R/W

23.2.2.9 Offset 0Eh: TWDS – TCO Watchdog Status Register

Table 696. Offset 0Eh: TWDS – TCO Watchdog Status Register

I/O Offset: *TCOBASE +0Eh*	***Size:*** *8 bit*
Default Value: *00h*	***Power Well:*** *Resume*

Bits	Name	Description	Reset Value	Access
07:00	TWDS	The value written to this register is passed via SMBus to an External LAN controller. It can be used by the BIOS or system management software to indicate more details on the boot progress. The register is reset to 00h based on a RSMRST# (but not PCI reset).	00h	RW

23.2.2.10 Offset 10h: LE – Legacy Elimination Register

Table 697. Offset 10h: LE – Legacy Elimination Register

I/O Offset: *TCOBASE +10h*	***Size:*** *8 bit*
Default Value: *11h*	***Power Well:*** *Core*

Bits	Name	Description	Reset Value	Access
07:02	Reserved	Reserved	00h	
01	IRQ12_CAUSE	0 = When software sets the bit to 0, IRQ12 is low (not asserted). 1 = When software sets the bit to 1, IRQ12 is high (asserted).	1	RW
00	IRQ1_CAUSE	0 = When software sets the bit to 0, IRQ1 is low (not asserted). 1 = When software sets the bit to 1, IRQ1 is high (asserted).	1	RW

23.2.2.11 Offset 12 - 13h: TTMR – TCO Timer Initial Value Register

Table 698. Offset 12h: TTMR – TCO_TMR Register

I/O Offset: *TCOBASE + (12 - 13h)*	***Size:*** *16 bits*
Default Value: *0004h*	***Power Well:*** *Core*

Bits	Name	Description	Reset Value	Access
15:10	Reserved	Reserved	00h	
09:00	TTMR	Value that is loaded into the timer each time the TCO_RLD register is written. Values of 0000h or 0001h are ignored and must not be attempted. The timer is clocked at approximately 0.6 s, and thus allows timeouts ranging from 1.2 s to 613.8 s. ***Note:*** The timer has an error of +/- 1 tick (0.6 s). The TCO Timer only counts down in the S0 state.	004h	RW

23.3 TCO Signal Usage

23.3.1 INTRUDER# Signal

This signal can be used to detect the chassis being opened. The activation of this signal can be used to cause an SMI#, and is reported via the event mechanism. If SMI# is desired, the signals level can be read, so this can be used as a type of general purpose input.

23.3.2 Pin Straps

Some of the TCO functions are decided at power up (rising edge of PWROK).

23.3.3 SMLINK Signals

The Intel® 3100 Chipset supports TCO compatible mode connectivity. TheIICHsupports External LAN controllers. An external LAN Controller can be used to receive or retrieve TCO message or information on Host SMBus if needed. In Legacy TCO mode messages are driven via SMLink.

For the Intel® 3100 Chipset, messages on this link use SMBus protocol at the rates described in Chapter 26.0, "Device 31, Function 3: SMBus Controller Functional Description," for TCO compatible mode.

23.4 TCO Theory of Operation

23.4.1 Overview

The system management functions are designed to allow the system to diagnose failing subsystems. The intent of this logic is that the system management functionality be provided without the aid of an external micro controller.

The Intel® 3100 Chipset's System Management logic allows for diagnostic and recovery software to be distributed between an SMI handler and operating system-based code.

23.4.2 Detecting a DOA CPU or System

When the processor is reset, it is expected to fetch its first instruction. If the processor fails to fetch the first instruction after reset, the TCO timer times out twice and Intel® 3100 Chipset asserts PLTRST#.

If TCO Reboots are not enabled, then:

1. The SMLink sends out the first eight bits of the message. After the eighth bit, the logic stalls because there is no integrated LAN controller to send the ACK. The logic then aborts the transfer. External logic monitors the toggling and use that to drive an LED.
2. If an External LAN controller is connected send the appropriate message to the External LAN controller.

If TCO Reboots are enabled, then the Intel® 3100 Chipset attempts to reboot the system.

1. If the NO-REBOOT bit (bit 5 of the GCS register, offset 3410-3413) is set (no reboots are intended)

and

SECOND_TO_STS bit (TCO I/O Offset 06h, bit 1) is set

and

DOACPU_STS bit (TCO I/O Offset 06h, bit 2) is set,

then the Intel® 3100 Chipset indicates this in the TCO message by setting the CPU Missing bit in the message.

2. If the NO-REBOOT bit (bit 5 of the GCS register, offset 3410-3413) is not set (reboots intended)

and

SECOND_TO_STS bit (TCO I/O Offset 06h, bit 1) is set,

then

the Intel® 3100 Chipset attempts to reboot. After the reboot, the SECOND_TO_STS bit is still set. If the CPU fails to fetch the first instruction, the DOA_CPU_STS bit is set, and when the TCO timer times out (actually for the third time, the first two caused the SECOND_TO_STS bit to be set), then the Intel® 3100 Chipset sets the CPU MISSING EVENT bit for the TCO message.

23.4.3 Handling an Operating System Lockup

Under some conditions, the operating system may lock up. To handle this, the TCO Timer is used with the following algorithm:

1. BIOS programs the TCO Timer, via the TCO_TMR register with an initial value.
2. An operating system-based software agent periodically writes to the TCO_RLD register to reload the timer and keep it from generating the SMI#. The software agent can read the TCO_RLD register to see if it is close to timing out, and possibly determine if the time-out should be increased. The operating system can also modify the values in the TCO_TMR register.
3. If the timer reaches 0, an SMI# can be generated. This should only occur if the operating system was not able to reload the timer. It is assumed that the operating system is not able to reload the timer if it has locked up.
4. Upon generating the SMI#, the TCO Timer automatically reloads with the default value of 04h and start counting down.
5. The SMI handler can then:
   a. Read the TIMEOUT bit in the TCO_STS register to check that the SMI# was caused by the TCO timer. The SMI handler **must** also clear the TIMEOUT bit.
   b. Write to the TCO_RLD register to reload the timer to make sure the TCO timer does not reach 0 again.
   c. Attempt to recover. May need to periodically reload the TCO timer.

The exact recovery algorithm is system-specific.

Note: If the SMI handler was not able to clear the TIMEOUT bit and write to the TCO_RLD register, the timer reaches zero a second time approximately 2.4 s later. At that point, the hardware is assumed to be locked up, and the timer reads zero a second time, which causes the SECOND_TO_STS bit to be set. At that point the logic resets the platform if the reboots are enabled.

23.4.4 Handling a CPU or Other Hardware Lockup

If after the TIMEOUT SMI is generated, and the TCO timer again reaches 0, and reboots are enabled, the System Management logic resets (and reboot) the system. This is in the case where the CPU or other system hardware is locked up. During every boot, BIOS must read the SECOND_TO_STS bit in the TCO_STS register to see if this is normal boot or a reboot due to the timeout. The BIOS may also check the OS_POLICY bits to see if it should try another boot or shutdown.

23.4.5 Handling an Intruder

The Intel® 3100 Chipset has an input signal, INTRUDER#, that can be attached to a switch that is activated by the system's case being open. This input has a two RTC clock debounce. If INTRUDER# goes active (after the debouncer), this sets the INTRD_DET bit in the TCO_STS register. INTRUDER# can go active in any power state.

The INTRD_SEL bits in the TCO_CNT register can enable the Intel® 3100 Chipset to cause an SMI# or TCO interrupt. The software can also directly read the status of the INTRUDER# signal (high or low) by clearing and then reading the INTRD_DET bit. This allows the signal to be used as a GPI if the intruder function is not required.

Note: The INTRD_DET bit resides in the Intel® 3100 Chipset's RTC well, and is set and cleared synchronously with the RTC clock. Thus, when software attempts to clear INTRD_DET (by writing a one to the bit location) there may be as much as two RTC clocks (about 65 µs) delay before the bit is actually cleared. Also, the INTRUDER# signal must be asserted for a minimum of 1 ms to guarantee that the INTRD_DET bit is set.

Note: If the INTRUDER# signal is still active when software attempts to clear the INTRD_DET bit, the bit remains set and the SMI is generated again immediately. The SMI handler can clear the INTRD_SEL bits to avoid further SMIs. However, if the INTRUDER# signal goes inactive and then active again, there is not further SMIs, since the INTRD_SEL bits would select that no SMI# be generated.

23.4.6 Handling a Potentially Failing Power Supply

It may be possible to detect that a power supply may fail in the near future by monitoring its voltages for fluctuations. To support such an application, external A/Ds with programmable thresholds could be included via SMBus/I^2C. Upon receiving the SMBus/I^2C message, the Intel® 3100 Chipset can generate an SMI or interrupt. The SMI handler (or operating system-based extension) could then attempt to send a message before the power completely fails.

Another option would be to build an A/D into the power supply itself. Another signal, other than PWROK, could report that the power supply might soon fail.

23.4.7 Handling an ECC Error or Other Memory Error

The IMCH provides a message to indicate that it would like to cause an SMI#, SCI, SERR#, or NMI. The software must check the IMCH as to the exact cause of the error.

23.4.8 SMM to Operating System and Operating System to SMM Calls

There may be interaction between an SMI handler and operating system-related code. Two 8-bit data registers are provided.

1. The SMI handler generates an interrupt to the operating system by writing to the TCO_DAT_OUT register. This sets the TCO_INT_STS bit in the TCO_STS register. The interrupt is cleared by writing a one to the TCO_INT_STS bit.
2. The operating system (or driver) can generate an SMI# by writing to the TCO_DAT_IN register. This sets the OS_SMI_STS bit in the TCO_STS register. The SMI# is cleared by writing a one to the OS_SMI_STS bit.

Reads to the TCO_DAT_IN and TCO_DAT_OUT register do not effect the SMI# or INTERRUPT.

Writing a one to the NMI_NOW bit allows for an immediate NMI.

23.4.9 Detecting an Improper FWH Programming

The Intel® 3100 Chipset can detect the case where the FWH is not programmed. This results in the first instruction fetched to have a value of FFh. If this occurs, the Intel® 3100 Chipset sets the BAD_BIOS bit.

23.4.10 IRQ1 and IRQ12 for Legacy Elimination

The new IRQ1 and IRQ12 sources are each logically ANDed with the respective IRQ1 and IRQ12 that come from the SERIRQ logic. This is necessary because the SERIRQ logic reports IRQ1 and IRQ12 to be high (active), since there is no Super I/O to drive them low.

Note: In a system that does have a Super I/O, the new bits must be left at one, since it is ANDed with the Super I/Os IRQ. **Do not attempt to write these bits to 0 in a system that has a keyboard controller (such as in a Super I/O).** It is not validated, and is highly likely to cause errors.

The following algorithm assumes the byte is being sent from the keyboard. The byte being sent from the mouse is equivalent. The setup to the area of interest is left at a high level in this description. The area of interest is then described in more detail.

1. An SMI is received and discovered to be a USB interrupt.
2. The interrupt is discovered to be due to a TD associated with a keyboard device.
3. The data is analyzed and it is determined that the interrupt is due to a new key press.
4. The USB key-code is translated into the equivalent scan code set 2 (SS2) PS/2 scan code.
5. The result is queued on a queue of data to be sent from the keyboard to the system.
6. Other USB interrupts are handled.
7. The keyboard controller emulation code is executed at exit. It determines if the conditions are correct to return a byte to the system (e.g., emulated OBF indicates empty, keyboard interface not disabled, etc.). If not, the emulation exits awaiting the next event.
8. The queue of data to be sent from the keyboard to the system is found to contain a byte to be returned.
9. Given the typical keyboard controller configuration, it is translated from SS2 to SS1.

End of Setup.

10. The byte to be returned is stored in the emulated output buffer.
11. The emulation does the out to the port to enable IRQ 1.
12. The emulation exits.
13. time passes
14. The system code services the interrupt and reads port 60h.
15. The UHCI traps the read and causes an SMI trap.
16. The trap is determined to be caused by the read from port 60h (TBY60R in LEGSUP).
17. The emulation code clears the interrupt register thus turning off IRQ 1 and / or IRQ 12.

Note: The emulation code can validly do this each time there is a read from 60h since that is what the keyboard controller would do as well. (There is no time where both IRQ 1 and IRQ 12 must be asserted simultaneously by either the kbc or the emulated kbc. This would be a violation of the kbc/system protocol.)

23.5 Event Reporting via SMLink/SMBus

The Intel® 3100 Chipset has SMLINK signals to support TCO compatible mode. Event reporting is accomplished via the SMLINK signals.

23.5.1 Overview

23.5.1.1 TCO Compatible Mode

The Intel® 3100 Chipset can function directly with a LAN controller to report message to a network management console without the aid of the system CPU. This is crucial in cases where the CPU is malfunctioning or cannot function due to being in a low-power state.

The basic scheme is to send specific messages via the SMLink Interface to the LAN. Upon receiving the SMLink message, the LAN has a prepared ethernet message that it can send to a network management console. The prepared message is stored in a non-volatile memory connected directly to the LAN.

Messages are sent by the Intel® 3100 Chipset to the LAN either because a specific event has occurred (see Table 699), or they are sent periodically. The event messages have exactly the same form.

Table 699. Event Transitions that Cause Messages

Event	Assertion	Deassertion	Comments
INTRUDER# pin	yes	no	
THRM# pin	yes	yes	The THRM# pin is isolated when the core power is off, thus preventing this event in S3,S5.
Watchdog Timer Expired	yes	no (NA)	
SEND_NOW bit	yes	NA	Occurs in G0
GPIO[11]/SMBALERT# pin	yes	yes	
BATLOW#	yes	yes	
CPU_PWR_FLR	yes	no	

Note: The GPIO[11]/SMBALERT# pin triggers an event message (when enabled by the GPIO11_ALERT_DISABLE bit) regardless of whether it is configured as a GPI or not.

Whenever an event occurs that causes the Intel® 3100 Chipset to send a new message, it increments the SEQ[03:00] field.

If a triggering event occurs while a message is already being generated and sent, the new event may not appear in the current message. If not, then a second message is generated, the SEQ[03:00] field increments to report the new event.

The following rules/steps apply if the system is in a G0 state and the policy is to reboot the system after a hardware lockup:

1. Upon detecting the lockup the SECOND_TO_STS bit is set. The Intel® 3100 Chipset may send up to one event message to the LAN. The Intel® 3100 Chipset then attempts to reboot the CPU.

2. If the reboot at step 1 is successful then the BIOS must clear the SECOND_TO_STS bit. This prevents any further messages from being sent. The BIOS may then perform addition recovery/boot steps.

Warning: It is important the BIOS clears the SECOND_TO_STS bit, as the alerts interfere with the LAN device driver from working properly. The alerts reset part of the LAN and would prevent an operating system's device driver from sending or receiving some messages.

3. If the reboot attempt in step 1 is not successful, then the timer timeouts a third time. At this point the system has locked up and was unsuccessful in rebooting. The Intel® 3100 Chipset does not attempt to automatically reboot again. The Intel® 3100 Chipset starts sending a message every period (30-32 seconds). This continue until some external intervention occurs (reset, power failure, etc.).
4. After step 3 (unsuccessful reboot after third timeout), if the user presses a Power Button Override, the system goes to an S5 state. The Intel® 3100 Chipset continues sending the messages every period.
5. After step 4 (power button override after unsuccessful reboot) if the user presses the Power Button again, the system must wake to an S0 state and the CPU must start executing the BIOS.
6. If step 5 (power button press) is successful in waking the system, The Intel® 3100 Chipset continues sending messages every period until the BIOS clears the SECOND_TO_STS bit.

Warning: It is important the BIOS clears the SECOND_TO_STS bit, as the messages interfere with the LAN device driver from working properly. The alerts reset part of the LAN and prevents an operating system's device driver from sending or receiving some LAN packets.

7. If step 5 (power button press) is unsuccessful in waking the system, The Intel® 3100 Chipset continues sending a message every period. The Intel® 3100 Chipset does not attempt to automatically reboot again. The Intel® 3100 Chipset starts sending a message every period (30-32 seconds). This continues until some external intervention occurs (reset, power failure, etc.).

Note: A system that has locked up and can not be restarted with the power button press is assumed to have broken hardware (bad power supply, short circuit on some bus, etc.), and is beyond the Intel® 3100 Chipset's recovery mechanisms.

8. After step 3 (unsuccessful reboot after third timeout), if a reset is attempted (using a button that pulses PWROK low or via the message on the SMBus Slave Interface), the Intel® 3100 Chipset attempts to reset the system.
9. After step 8 (reset attempt) if the reset is successful, then the BIOS is run. The Intel® 3100 Chipset continues sending a message every period until the BIOS clears the SECOND_TO_STS bit.

Warning: It is important the BIOS clears the SECOND_TO_STS bit, as the messages interfere with the LAN device driver from working properly. The alerts reset part of the LAN and prevent an operating system's device driver from sending or receiving some LAN packets.

10. After step 8 (reset attempt), if the reset is unsuccessful, then the Intel® 3100 Chipset continues sending a message every period. The Intel® 3100 Chipset does not attempt to reboot the system again without external intervention.

Note: A system that has locked up and can not be restarted with the power button press is assumed to have broken hardware (bad power supply, short circuit on some bus, etc.), and is beyond the Intel® 3100 Chipset's recovery mechanisms.

The following rules/steps apply if the system is in a G0 state and the policy is for the Intel® 3100 Chipset to **not** reboot the system after a hardware lockup:

1. Upon detecting the lockup the SECOND_TO_STS bit is set. The Intel® 3100 Chipset sends a message with the Watchdog (WD) Event status bit set (and any other bits that must also be set). This message is sent as soon as the lockup is detected, and is sent with the next (increment) sequence number.
2. After step 1, the Intel® 3100 Chipset sends a message every period until some external intervention occurs.
3. Rules/steps 4-10 apply if no user intervention (resets, power button presses, SMBus reset messages) occur after a third timeout of the watchdog timer. If the intervention occurs before the third timeout, then jump to step11.
4. After step 3 (third timeout), if the user does a Power Button Override, the system goes to an S5 state. The Intel® 3100 Chipset continues sending messages at this point.
5. After step 4 (power button override), if the user presses the power button again, the system must wake to an S0 state and the CPU must start executing the BIOS.
6. If step 5 (power button press) is successful in waking the system, the Intel® 3100 Chipset continues sending messages until the BIOS clears the SECOND_TO_STS bit.

Warning: It is important the BIOS clears the SECOND_TO_STS bit, as the alerts interfere with the LAN device driver from working properly. The alerts reset part of the LAN and would prevent an operating system's device driver from sending or receiving some messages.

7. If step 5 (power button press) is unsuccessful in waking the system, the Intel® 3100 Chipset continues sending messages. The Intel® 3100 Chipset does not attempt to reboot the system again until some external intervention occurs (reset, power failure, etc.).

Note: A system that has locked up and can not be restarted with power button press is probably very broken (bad power supply, short circuit on some bus, etc.) and beyond the Intel® 3100 Chipset's recovery mechanisms.

8. After step 3 (third timeout), if a reset is attempted (using a button that pulses PWROK low or via the message on the SMBus Slave Interface), the Intel® 3100 Chipset attempts to reset the system.
9. If step 8 (reset attempt) is successful, then the BIOS is run. The Intel® 3100 Chipset continues sending messages until the BIOS clears the SECOND_TO_STS bit.

Warning: It is important the BIOS clears the SECOND_TO_STS bit, as the alerts interfere with the LAN device driver from working properly. The alerts reset part of the LAN and would prevent an operating system's device driver from sending or receiving some messages.

10. If step 8 (reset attempt), is unsuccessful, then the Intel® 3100 Chipset continues sending messages. The Intel® 3100 Chipset does not attempt to reboot the system again without external intervention.

Note: A system that has locked up and can not be restarted with the power button press is broken (bad power supply, short circuit on some bus, etc.)

11. This and the following rules/steps apply if the user intervention (power button press, reset, SMBus message, etc.) occur prior to the third timeout of the watchdog timer.
12. After step 1 (second timeout), if the user does a Power Button Override, the system goes to an S5 state. The Intel® 3100 Chipset continues sending messages at this point.
13. After step 12 (power button override), if the user presses the power button again, the system must wake to an S0 state and the CPU must start executing the BIOS.

14. If step 13 (power button press) is successful in waking the system, the Intel® 3100 Chipset continues sending messages until the BIOS clears the SECOND_TO_STS bit.

Warning: It is important the BIOS clears the SECOND_TO_STS bit, as the alerts interfere with the LAN device driver from working properly. The alerts reset part of the LAN and would prevent an operating system's device driver from sending or receiving some messages.

15. If step 13 (power button press) is unsuccessful in waking the system, the Intel® 3100 Chipset continues sending messages. The Intel® 3100 Chipset does not attempt to reboot the system again until some external intervention occurs (reset, power failure, etc.).

Note: A system that has locked up and can not be restarted with power button press is broken (bad power supply, short circuit on some bus, etc.) and beyond the Intel® 3100 Chipset's recovery mechanisms.

16. After step 1 (second timeout), if a reset is attempted (using a button that pulses PWROK low or via the message on the SMBus Slave Interface), the Intel® 3100 Chipset attempts to reset the system.
17. If step 16 (reset attempt) is successful, then the BIOS is run. The Intel® 3100 Chipset continues sending messages until the BIOS clears the SECOND_TO_STS bit.

Warning: It is important the BIOS clears the SECOND_TO_STS bit, as the alerts interfere with the LAN device driver from working properly. The alerts reset part of the LAN and would prevent an operating system's device driver from sending or receiving some messages.

18. If step 16 (reset attempt), is unsuccessful, then the Intel® 3100 Chipset continues sending messages. The Intel® 3100 Chipset does not attempt to reboot the system again without external intervention.

Note: A system that has locked up and can not be restarted with power button press is broken (bad power supply, short circuit on some bus, etc.)

The following rules apply if the system is in a G1 (S3) state:

- The Intel® 3100 Chipset sends a message every period (30-32 seconds).
  - ..... If an event occurs prior to the system being shut down, the Intel® 3100 Chipset immediately sends another event message with the next (incremented) sequence number.
  - ..... After the event, it resumes sending messages.

Note: There is a boundary condition when a hardware event happens right as the system transitions into a G0 state. In this condition, the hardware sends messages even though the system is in a G0 state (and the status bits could potentially indicate that). Normally the IICH does not send messages in the G0 state (except in the case of a lockup).

Note: A spurious alert could occur in the following sequence:

a. The CPU has initiated an alert using the SEND_NOW bit
b. During the alert, the THRM#, INTRUDER# or GPI[11] changes state
c. The system then goes to a non-S0 state

Once the system transitions to the non-S0 state, it may send a single alert with an incremented SEQUENCE number.

Note: An inaccurate alert message can be generated in the following scenario:

a. The system successfully boots after a second watchdog Timeout occurs.

b. PWROK goes low (typically due to a reset button press) or a power button override occurs (before the SECOND_TO_STS bit is cleared).

c. An alert message indicating that the CPU is missing or locked up is generated with a new sequence number.

23.5.2 Message Format

The Event message is an SMBus Block Write sent to the External LAN Controller's SMBus address, as shown in Table 700.

Table 700. SMBus Message Format

S																															
1																															
Address Byte									**OpCode Byte**									**Length Byte**													
Address							**D**		**Opcode**									**Length**													
1	**1**	**0**	**0**	**1**	**0**	**0**	**0**		**1**		**0**	**0**	**1**	**0**	**0**	**0**		**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**						
Data Byte 1									**Data Byte 2**									**Data Byte 3**													
(See Description)									**(See Description)**									**Sequence**				PwrSt		Rsvd							
CT	**BT**	**CS**	**Wt**	**SE**	**PE**	**FE**	**G**		**BB**	**BL**	**CF**	**0**	**0**	**0**	**0**	**0**		**B3**	B2	B1	B0	B1	B0	0	0						
Data Byte 4									**Data Byte 5**									**Data Byte 6**													
Message 1 Register									**Message 2 Register**									**WD Status Register**													
B7	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**		**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**		**B7**	B6	B5	B4	B3	B2	B1	B0						
Data Byte 7									**Data Byte 8**																						
Reserved									**Reserved**									**P**													
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**		**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		**1**													

Table 701. Message Address Byte

Field	Bit Length	Comment
Start	1	'1' to indicate the start of a packet
Address	7	LAN SMBus Address. Is always 1100100.
Dir	1	'0' to indicate write cycle
Ack	1	Returned by External LAN Controller

Note: For the System Power State field: 00 = G0, 01 = G1, 10 = G2, 11=PreBoot. The preboot state is entered when the SLP_S3#, SLP_S4# and SLP_S5# signals go from low to high. The indication switches to the G0 state when CPURESET Done ACK completion packet sent to the IMCH. This corresponds to the time when the CPU has been reset. If the CPU is locked up, then the CPU EVENT bit is set.

23.5.3 Connecting an External LAN Controller

The Intel® 3100 Chipset's TCO logic sends the message on the SMLINK signals in TCO compatible mode. An External LAN Controller claims these cycles.

When sending the messages to the external LAN Controller, the Intel® 3100 Chipset's IICH abides by the standard SMBus rules associated with collision detection. It delays starting a message until the link is idle, and detects collisions. If a collision is detected, the Intel® 3100 Chipset drops that message.

24.0 Device 31, Function 0: General Purpose I/O

24.1 Overview

The GPIO signals are divided into several types, as shown in Table 702.

Table 702. GPIO Signal Types

Type	Intel® 3100 Chipset Allocation (Only the GPIOs listed below are implemented)
General Purpose Input on Core Well	Input GPIO[00:07,12:13] Can cause a wake event, SMI#, or SCI.
General Purpose Input on Resume Well	Input GPIO[08:11,14:15] Can cause a wake event, SMI#, or SCI.
General Purpose Output on Core Well	Output GPIO[16:21,23]
General Purpose Input/Output on Resume Well	GPIO[24:25]
General Purpose Input on Core Well	Input GPIO[26]
General Purpose Input/Output on Resume Well	GPIO[27:28]
General Purpose Input on Core Well	Input GPIO[29:31]
General Purpose Input/Output on Core Well	GPIO[32:39]
General Purpose Input on Core/CPU I/O Well	Input GPIO[40:47]
General Purpose Output on Core Well	Output GPIO[48:55]

24.2 General Purpose I/O Registers (D31, F0)

The control for the general purpose I/O signals is handled through separate 64-byte I/O space. The base offset for this space is selected by the GPIO_BAR register in D31, F0 configuration space.

Note: For the following registers, if a bit is allocated for a GPIO that doesn't exist, unless otherwise indicated, the bit always reads 0 and values written to that bit have no effect.

24.2.1 GPIO Register Address Map

Note: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Table 703. GPIO Register Summary Table

GPIOBASE + Offset		Symbol	Register Name/Description	Default	Type
Start	End				
General Registers					
00h	03h	GPIO_USE_SEL	GPIO Use Select Register	1B0C01C0h	RW
04h	07h	GP_IO_SEL	GPIO Input/Output Select Register	E400 FFFFh	RW
0Ch	0Fh	GP_LVL	GPIO Level for Input or Output Register	FF3F0000h	RW
Output Control Registers					
18h	1Bh	GPO_BLINK	GPIO Blink Enable Register	0004 0000h	RW
Input Control Registers					
2Ch	2Fh	GPI_INV	GPIO Signal Invert Register	00000000h	RW
30h	33h	GPIO_USE_SEL2	GPIO Use Select 2 [63:32] Register	00000006h	RW
34h	37h	GP_IO_SEL2	GPIO Input/Output Select 2 [63:32] Register	00000300h	RW
38h	3Bh	GP_LVL2	GPIO Level for Input or Output 2 [63:32] Register	00030207h	RW

Note: The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Note: Reserved bits are Read Only.

24.2.2 Register Descriptions

24.2.2.1 Offset 00 - 03h: GPIO_USE_SEL1– GPIO Use Select 1 [31:0] Register

Table 704. Offset 00 - 03h: GPIO_USE_SEL1– GPIO Use Select 1 [31:0] Register

I/O Address: *GPIOBASE +00 - 03h*	***Size:*** *32 bit*
Default Value: *1B0C01C0h*	***Power Well:*** *Core for 0:7,12:13, 16:21,23,26,29:31* *Resume for 8:11,14:15, 25,27:28*

Bits	Name	Description	Reset Value	Access
31:29, 26,23, 21:20, 15:14, 13:09, 05:00	GPIO_USE_SEL	Enables GPIO[n] (where n is the bit number) to be used as a GPIO, rather than for the native function. 1 = Signal used as GPIO (or unmuxed). 0 = Signal used as native function. Example: software sets bit 2 in this register to enable GPIO[2] (instead of using that signal for PIRQ[E]). ***Notes:*** 1. Bit 22 is not supported because there is no corresponding GPIO. 2. The following bits are always 1 because they are unmuxed: 06, 07, 08, 18, 19, 24:25, 27:28. 3. Bit 16 is not supported because the GPIO selection is controlled by Bit 0 (REQ/GNT pair). 4. Bit 17 is not supported because the GPIO selection is controlled by Bit 1 (REQ/GNT pair). 5. Bits 0, 1, 9, 10, 14:15 **MUST BE PROGRAMMED TO 1 by BIOS**. 6. If GPIO[n] does not exist, then the bit in this register is always read as 0 and writes have no effect. 7. After a full reset (RSMRST#) all multiplexed signals in the resume and core wells are configured as their native function rather than as a GPIO. After just a PLTRST#, the GPIOs in the core wells are configured as their native function. 8. When configured to GPIO mode, the muxing logic presents the inactive state to native logic that uses the pin as an input.	1B0C01C0h	RW

24.2.2.2 Offset 04 - 07h: GP_IO_SEL1 – GPIO Input/Output Select 1 [31:0] Register

Table 705. Offset 04 - 07h: GP_IO_SEL1 – GPIO Input/Output Select 1 [31:0] Register

I/O Address: *GPIOBASE +04 - 07h*	***Size:*** *32 bit*
Default Value: *E400FFFFh*	***Power Well:*** *See below*

Bits	Name	Description	Reset Value	Access
31:29	GPI[31:29]	Always 1. The GPI pins are always inputs.	111b	RO
28:27	GP_IO_SEL[28:27]	0 = The GPIO signal is programmed as an output. 1 = The corresponding GPIO signal (if enabled in the GPIO_USE_SEL register) is programmed as an input.	00b	RW
26	GPI[26]	Always 1. The GPI pins are always inputs.	1b	RO

Table 705. Offset 04 - 07h: GP_IO_SEL1 – GPIO Input/Output Select 1 [31:0] Register

I/O Address: *GPIOBASE +04 - 07h*	***Size:*** *32 bit*
Default Value: *E400FFFFh*	***Power Well:*** *See below*

Bits	Name	Description	Reset Value	Access
25:24	GP_IO_ SEL[25:24]	0 = The GPIO signal is programmed as an output. 1 = The corresponding GPIO signal (if enabled in the GPIO_USE_SEL register) is programmed as an input.	00b	RW
23:16	GPO[23:16]	**Always 0.** The GPO pins are always outputs.	00h	RO
15:00	GPI[15:0]	**Always 1.** The GPI pins are always inputs.	FFFFh	RO

24.2.2.3 Offset 0C - 0Fh: GP_LVL1 – GPIO Level 1 for Input or Output [31:0] Register

Table 706. Offset 0C - 0Fh: GP_LVL1 – GPIO Level 1 for Input or Output [31:0] Register (Sheet 1 of 2)

I/O Address: *GPIOBASE +0C - 0Fh*	***Size:*** *32 bit*
Default Value: *FF3F0000h*	***Power Well:*** *See below*

Bits	Name	Description	Reset Value	Access
31:29	GP_LVL[31:29]	These bits correspond to input-only GPIO in the core well. The corresponding GP_LVL bit reflects the state of the input signal. Writes to these bits have no effect. 0 = Low 1 = High These bits correspond to GPIO that are in the core well and are reset to their native function by PLTRST#.	111b	RW
28:27	GP_LVL[28:27]	If GPIO[n] is programmed to be an output (via the corresponding bit in the GP_IO_SEL register), then the corresponding GP_LVL[n] bit can be updated by software to drive a high or low value on the output pin. If GPIO[n] is programmed as an input, then the corresponding GP_LVL bit reflects the state of the input signal. Writes have no effect. 0 = Low 1 = High These bits correspond to GPIO that are in the Resume well and are reset to their native function by RSMRST# and by a writing to the CF9h register.	11b	RW
26	GP_LVL[26]	This bit corresponds to input-only GPI in the core well. The corresponding GP_LVL bit reflects the state of the input signal. Writes to this bit have no effect. 0 = Low 1 = High This bit corresponds to a GPI that is in the core well and is reset to its native function by PLTRST#.	1b	RW
25:24	GP_LVL[25:24]	If GPIO[n] is programmed to be an output (via the corresponding bit in the GP_IO_SEL register), then the corresponding GP_LVL[n] bit can be updated by software to drive a high or low value on the output pin. If GPIO[n] is programmed as an input, then the corresponding GP_LVL bit reflects the state of the input signal. Writes have no effect. 0 = Low 1 = High These bits correspond to GPIO that are in the Resume well and are reset to their native function by RSMRST# and by a writing to the CF9h register.	11b	RW

Table 706. Offset 0C - 0Fh: GP_LVL1 – GPIO Level 1 for Input or Output [31:0] Register (Sheet 2 of 2)

I/O Address: GPIOBASE +0C - 0Fh			**Size:** 32 bit	
Default Value: FF3F0000h			**Power Well:** See below	
Bits	**Name**	**Description**	**Reset Value**	**Access**
23:16	GP_LVL[23:16]	These bits can be updated by software to drive a high or low value on the output pin. 0 = Low 1 = High These bits correspond to GPIOs that are in the core well and are reset to their native function by PLTRST#.	3Fh	RW
15:00	GP_LVL[15:00]	Reserved.	0000h	

24.2.2.4 Offset 18 - 1Bh: GPO_BLINK – GPIO Blink Enable Register

Table 707. Offset 18 - 1Bh: GPO_BLINK – GPIO Blink Enable Register

I/O Address: GPIOBASE +18 - 1Bh			**Size:** 32 bit	
Default Value: 0004 0000h			**Power Well:** See below	
Bits	**Name**	**Description**	**Reset Value**	**Access**
28:27,25	GPO_BLINK[28:27,25]	The setting of this bit has no effect if the corresponding GPIO signal is programmed as an input. 0 = The corresponding GPIO functions normally. 1 = If the corresponding GPIO is programmed as an output, the output signal blinks at a rate of approximately once per second. The high and low times have approximately 0.5 seconds each. The GP_LVL bit is not altered when this bit is set. The usage model for a blinking output is to control an LED. This value does not need to have exactly one second granularity, but must be close. The value of the corresponding GP_LVL bit remains unchanged during the blink process, and does not effect the blink in any way. The GP_LVL bit is not altered when programmed to blink. It remains at its previous value. These bits correspond to GPIO in the Resume well and are reset to their native function by RSMRST# or a write to the CF9h register or any other PLTRST#.	000h	RW
19:18	GPO_BLINK[19:18]	The setting of this bit has no effect if the corresponding GPIO signal is programmed as an input. 0 = The corresponding GPIO functions normally. 1 = If the corresponding GPIO is programmed as an output, the output signal blinks at a rate of approximately once per second. The high and low times are approximately 0.5 seconds each. The GP_LVL bit is not altered when this bit is set. The usage model for a blinking output is to control an LED. This value does not need to have exactly one second granularity, but must be close. The value of the corresponding GP_LVL bit remains unchanged during the blink process, and does not effect the blink in any way. The GP_LVL bit is not altered when programmed to blink. It remains at its previous value. These bits correspond to GPIO in the core well and are reset to their native function by PLTRST#.	01b	RW

24.2.2.5 Offset 2C - 2Fh: GPI_INV – GPIO Signal Invert Register

Table 708. Offset 2C - 2Fh: GPI_INV – GPIO Signal Invert Register

I/O Address: *GPIOBASE +2C - 2Fh*			***Size:*** *32 bit*	
Default Value: *00000000h*			***Power Well:*** *Core 7:0, 12, Resume 15:13, 11:8*	
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:16	Reserved	Reserved.	00h	
15:14, 11:08	GPI_ INV[15:14,11:8]	**Input Inversion**: These bits only have effect if the corresponding GPIO is used as an input. This is used to allow active-low and active-high inputs to cause SMI# or SCI. 0 = No bit is inverted. 1 = The corresponding data value in the GP_LVL bit is inverted. These bits correspond to GPI in the resume well and are reset to their native function by RSMRST# or a write to the CF9h register or any other PLTRST#. In the S0 state, the input signal must be active for at least two PCI clocks to ensure detection by the IICH. In the S3 or S5 states the input signal must be active for at least two RTC clocks to ensure detection. The setting of these bits have no effect if the corresponding GPIO is programmed as an output.	00b 0000b	RW
13:12	GPI_INV[13:12]	**Input Inversion**: These bits only has effect if the corresponding GPIO is used as an input. This is used to allow active-low and active-high inputs to cause SMI# or SCI. 0 = No bit is inverted. 1 = The corresponding data value in the GP_LVL bit is inverted. These bits correspond to GPI in the core well and are reset to their native function by PLTRST#.	00b	RW
07:00	GPI_INV[7:0]	**Input Inversion**: This bit only has effect if the corresponding GPIO is used as an input. This is used to allow active-low and active-high inputs to cause SMI# or SCI. 0 = No bit is inverted. 1 = The corresponding data value in the GP_LVL bit is inverted. These bits correspond to GPI in the core well and are reset to their native function by PLTRST#.	00h	RW

24.2.2.6 Offset 30 - 33h: GPIO_USE_SEL2 – GPIO Use Select 2 [63:32] Register

Table 709. Offset 30 - 33h: GPIO_USE_SEL2 – GPIO Use Select 2 [63:32] Register

I/O Address: GPIOBASE +30 - 33h ***Size:*** 32 bit

Default Value: 00000006h ***Power Well:*** CPU I/O for 17, Core for 16:00

Bits	Name	Description	Reset Value	Access
17, 09: 00	GPIO_ USE_SEL [49, 41: 32]	Enables GPIO[n] (where n is the bit number) to be used as a GPIO, rather than for the native function. Since these pins may be used as outputs for controlling power planes, switching the pin from functional to GPO mode must be glitch-free. 0 = Signal used as native function. 1 = Signal used as GPIO. After a full reset (RSMRST#) all multiplexed signals in the resume and core wells are configured as their native function rather than as a GPIO. After just a PLTRST#, the GPIO in the core well are configured as GPIO. ***Notes:*** 1. The following bits are not supported because there are no corresponding GPIOs: 03:07, 10:15, 18:31. 2. The following bits are always 1 because they are unmuxed: 01:02 3. Bit 16 is not supported. 4. Bit 8 **MUST BE PROGRAMMED TO 1 by BIOS**. 5. If GPIO[n] does not exist, then the bit in this register always reads as 0 and writes have no effect.	00000006h	RW

24.2.2.7 Offset 34 - 37h: GP_IO_SEL2 – GPIO Input/Output Select 2 [63:32] Register

Table 710. Offset 34 - 37h: GP_IO_SEL2 – GPIO Input/Output Select2 [63:32] Register

I/O Address: *GPIOBASE +34 - 37h* ***Size:*** *32 bit*

Default Value: *00000300h* ***Power Well:*** *CPU I/O for 17, Core for 16:0*

Bits	Name	Description	Reset Value	Access
31:18	GP_IO_SEL[63:50]	Always 0. No corresponding GPIO.	0	RO
17:16	GP_IO_SEL[49:48]	Always 0. These pins are always outputs.	0	RO
15:10	GP_IO_SEL[47:42]	Always 0. No corresponding GPIO.	0	RO
09:08	GP_IO_SEL[41:40]	Always 1. These pins are always inputs.	11b	RO
07:03	GP_IO_SEL[39:35]	Always 0. No corresponding GPIO.	0	RO
02:00	GP_IO_ SEL[34:32]	0 = GPIO signal is programmed as an output. 1 = Corresponding GPIO signal (if enabled in the GPIO_USE_SEL2 register) is programmed as an input.	000b	RW

24.2.2.8 Offset 38 - 3Bh: GP_LVL2 – GPIO Level for Input or Output 2 [63:32] Register

Table 711. Offset 38 - 3Bh: GP_LVL2 – GPIO Level for Input or Output 2 [63:32] Register

I/O Address: GPIOBASE +38- 3Bh	***Size:*** 32 bit
Default Value: 00030207h	***Power Well:*** CPU I/O for 17, Core for 16:00

Bits	Name	Description	Reset Value	Access
31:18	Reserved	Read-Only 0.	0	
17:16	GP_LVL[49:48]	The corresponding GP_LVL[n] bit can be updated by software to drive a high or low value on the output pin. 0 = low 1 = high Since these bits correspond to GPIO that are in the processor I/O and core well, respectively, these bits are reset to their native function by PLTRST#.	11b	RW
15:10	Reserved	Read-Only 0.	0	
09:08	GP_LVL[41:40]	The corresponding GP_LVL[n] bit reflects the state of the input signal. Writes have no effect. 0 = low 1 = high Since these bits correspond to GPIO that are in the core well they are reset to their native function by PLTRST#.	10b	RW
07:03	Reserved	Read-Only 0.	0	
02:00	GP_LVL[34:32]	If GPIO[n] is programmed to be an output (via the corresponding bit in the GP_IO_SEL register), then the corresponding GP_LVL[n] bit can be updated by software to drive a high or low value on the output pin. If GPIO[n] is programmed as an input, then the corresponding GP_LVL bit reflects the state of the input signal (1 = high, 0 = low). Writes have no effect. 0 = low 1 = high Since these bits correspond to GPIO that are in the core well and are reset to their native function by PLTRST#.	111b	RW

24.3 Additional GPIO Theory of Operation

24.3.1 Power Wells

Some GPIOs exist in the resume power plane. Care must be taken to make sure GPIO signals are not driven high into powered-down planes.

Some Intel® 3100 Chipset GPIOs may be connected to pins on devices that exist in the core well. If these GPIOs are outputs, there is a danger that a loss of core power (PWROK low) or a Power Button Override event results in the Intel® 3100 Chipset driving a pin to a logic ‘1’ to another device that is powered down.

24.3.2 SMI# and SCI Routing

The routing bits for GPIO[0:15] allow an input to be routed to SMI# or SCI, or neither. See Chapter 22.0, “Power Management” for the routing register. A bit can be routed to either an SMI# or an SCI, but not both.

24.3.3 Triggering

GPIO[0:15] have “sticky” bits on the input. See Chapter 22.0, “Power Management” for the GPE0_STS register and the ALT_GPI_SMI_STS register. As long as the signal goes active for at least two clocks, the Intel® 3100 Chipset keeps the sticky status bit active. The active level (high or low) can be selected via the GP_INV register.

If the system is in an S0 state, the GPI are sampled at 33 MHz, so the signal only needs to be active for about 60 ns to be latched. In S3,S5 states, the GPI are sampled at 32.768 kHz, and thus must be active for at least 61 µs to be latched.

Note: GPIs that are in the core well are not capable of waking the system from sleep states where the core well is not powered.

If the input signal is still active when the latch is cleared, it is again set (another edge is not required). This makes these signals “level” triggered inputs.

24.4 GPIO Summary Table

Table 712. GPIO Summary Table (Sheet 1 of 2)

GPIO#	PowerWell	Muxed	GPIO_USE_SEL	GP_IO_SEL	GP_LVL	GPO_BLINK	GPI_INV
0	Core	No	Yes – D=0	Always 1	Always 0	Always 0	Yes – D=0
1	Core	No	Yes – D=0	Always 1	Always 0	Always 0	Yes – D=0
2	Core	Yes	Yes – D=0	Always 1	Always 0	Always 0	Yes – D=0
3	Core	Yes	Yes – D=0	Always 1	Always 0	Always 0	Yes – D=0
4	Core	Yes	Yes – D=0	Always 1	Always 0	Always 0	Yes – D=0
5	Core	Yes	Yes – D=0	Always 1	Always 0	Always 0	Yes – D=0
6	Core	No	Always 1	Always 1	Always 0	Always 0	Yes – D=0
7	Core	No	Always 1	Always 1	Always 0	Always 0	Yes – D=0
8	Resume	No	Always 1	Always 1	Always 0	Always 0	Yes – D=0
9	Resume	No	Yes – D=0	Always 1	Always 0	Always 0	Yes – D=0
10	Resume	No	Yes – D=0	Always 1	Always 0	Always 0	Yes – D=0
11	Resume	Yes	Yes – D=0	Always 1	Always 0	Always 0	Yes – D=0
12	Core	No	Yes – D=0	Always 1	Always 0	Always 0	Yes – D=0
13	Core	No	Yes – D=0	Always 1	Always 0	Always 0	Yes – D=0
14	Resume	No	Yes – D=0	Always 1	Always 0	Always 0	Yes – D=0
15	Resume	No	Yes – D=0	Always 1	Always 0	Always 0	Yes – D=0
16	Core	No	Always 0	Always 0	Yes – D=1	Always 0	Always 0
17	Core	No	Always 0	Always 0	Yes – D=1	Always 0	Always 0
18	Core	No	Always 1	Always 0	Yes – D=1	Yes – D=1	Always 0
19	Core	No	Always 1	Always 0	Yes – D=1	Yes – D=0	Always 0
20	Core	No	Always 0	Always 0	Yes – D=1	Always 0	Always 0
21	Core	No	Yes – D=0	Always 0	Yes – D=1	Always 0	Always 0
22	Unimplemented		Always 0	Always 0	Always 0	Always 0	Always 0
23	Core	No	Yes – D=0	Always 0	Yes – D=0	Always 0	Always 0
24	Resume	No	Always 1	Yes – D=0	Yes – D=1	Always 0	Always 0
25	Resume	No	Always 1	Yes – D=0	Yes – D=1	Yes – D=0	Always 0

Table 712. GPIO Summary Table (Sheet 2 of 2)

GPIO#	PowerWell	Muxed	GPIO_USE_SEL	GP_IO_SEL	GP_LVL	GPO_BLINK	GPI_INV
26	Core	No	Yes – D=0	Always 1	Yes – D=1	Always 0	Always 0
27:28	Resume	No	Always 1	Yes – D=0	Yes – D=1	Yes – D=0	Always 0
29:31	Core	No	Yes – D=0	Always 1	Yes – D=1	Always 0	Always 0
32	Core	No	Yes – D=0	Yes – D=0	Yes – D=1	Always 0	Always 0
33:34	Core	No	Always 1	Yes – D=0	Yes – D=1	Always 0	Always 0
35:39	Unimplemented		Always 0	Always 0	Always 0	Always 0	Always 0
40	Core	No	Yes – D=0	Always 1	Yes – D=0	Always 0	Always 0
41	Core	Yes	Yes – D=0	Always 1	Yes – D=1	Always 0	Always 0
42:47	Unimplemented		Always 0	Always 0	Always 0	Always 0	Always 0
48	Core	No	Always 0	Always 0	Yes – D=1	Always 0	Always 0
49	CPU I/O	Yes	Yes – D=0	Always 0	Yes – D=1	Always 0	Always 0
50:55	Unimplemented		Always 0	Always 0	Always 0	Always 0	Always 0

25.0 Device 29, Functions 0, 1: USB(1.1) Controllers

25.1 USB1 Configuration Registers

Warning: Register address locations that are not shown in Table 713 must be treated as Reserved. Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 713. USB1 Configuration Registers

Start	End	Symbol	Register Name / Function	Function 0 Default	Function 1 Default	Access
00h	03h		Identifiers Register	2688h	2689h	RO
04h	05h		Command Register	0000h	0000h	RW
06h	07h		Device Status Register	0280h	0280h	RW
08h	08h		Revision ID Register	See Desc	See Desc	RO
09h	09h		Programming Interface Register	00h	00h	RO
0Ah	0Ah		Sub Class Code Register	03h	03h	RO
0Bh	0Bh		Base Class Code Register	0Ch	0Ch	RO
0Dh	0Dh		Master Latency Timer Register	00h	00h	RO
0Eh	0Eh		Header Type Register	see desc.	00h	RO
20h	23h		Base Address Register	00000001h	00000001h	RW
2Ch	2Dh	USBx_ SVID	Subsystem Vendor ID Register	0000h	0000h	RWO
2Eh	2Fh	USBx_ SID	Subsystem ID Register	0000h	0000h	RWO
3Ch	3Ch		Interrupt Line Register	00h	00h	RW
3Dh	3Dh		Interrupt Pin Register	see desc.	see desc.	RO
60h	60h		USB Release Number Register	10h	10h	RO
C0h	C1h		USB Legacy Keyboard Enable Register	2000h	2000h	RW
C4h	C4h		USB Resume Enable Register	00h	00h	RW
C8h	C8h		USB Core Well Policy Register	00h	00h	RW
F8h	FBh		Manufacturer's ID Register	00010F80h	00010F80h	

25.1.1 Register Details

25.1.1.1 Offset 00 - 03h: Identifiers Register

Table 714. Offset 00 - 03h: Identifiers Register

Device: 29	***Function:*** 0, 1
Offset: 00 -03h	***Size:*** 32 bit
Default Value: See Desc	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:16	DID	**Device ID:** This 16-bit field is defined as follows: Function 0 2688h Function 1 2689h	See Desc	RO
15:00	VID	**Vendor ID:** 16-bit field which indicates the company vendor as Intel.	8086h	RO

25.1.1.2 Offset 04 - 05h: Command Register

Table 715. Offset 04 - 05h: Command Register

Device: 29	***Function:*** 0, 1
Offset: 04 - 05h	***Size:*** 16 bit
Default Value: 0000h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:11	Reserved	Reserved	0	
10	INTDIS	**Interrupt Disable:** 0 = Enable. The function is able to generate its interrupt to the interrupt controller. 1 = Disable. The function is not capable of generating interrupts. The corresponding Interrupt Status bit is not affected by the interrupt enable.	0	R/W
09	FBE	**Fast Back to Back Enable:** Hardwired to'0'.	0	
08	SERREN	**SERR# Enable:** Hardwired to'0'.	0	
07	WCC	**Wait Cycle Control:** Hardwired to'0'.	0	
06	PER	**Parity Error Response:** Hardwired to'0'.	0	
05	VGAPS	**VGA Palette Snoop:** Hardwired to'0'.	0	
04	PMWE	**Postable Memory Write Enable:** Hardwired to'0'.	0	
03	SCE	**Special Cycle Enable:** Hardwired to'0'.	0	
02	BME	**Bus Master Enable:** 0 = Disable. 1 = Enable. Can generate cycles to main memory as a master for USB transfers.	0	R/W
01	MSE	**Memory Space Enable:** Hardwired to'0'.	0	
00	IOSE	**I/O Space Enable:** This bit controls access to the I/O space registers. 0 = Disable: Accesses to the USB I/O registers is Disabled. 1 = Accesses to the USB I/O registers is enabled. The Base Address register for USB must be programmed before this bit is set.	0	R/W

25.1.1.3 Offset 06 - 07h: Device Status Register

Table 716. Offset 06 - 07h: Device Status Register

Device: 29	***Function:*** 0, 1
Offset: 06 - 07h	***Size:*** 16 bit
Default Value: 0280h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15	DPE	**Detected Parity Error:** 0 = No parity error detected. 1 = Set when a data parity error is detected on writes to the UHCI register space or on read completions returned to the host controller. ***Note:*** Software sets this bit to 0 by writing a 1 to this bit location.	0	RWC
14	SSE	Reserved	0	
13	RMA	**Received Master-Abort Status:** 0 = No master abort generated by USB. 1 = USB as a master, receives a master abort. ***Note:*** Software sets this bit to 0 by writing a 1 to this bit location.	0	RWC
12	Reserved	Reserved	0	
11	STA	**Signaled Target-Abort Status:** 0 = Did not terminate a transaction to a USB function with a target abort. 1 = USB function is targeted with a transaction that terminates with a target abort. ***Note:*** Software sets this bit to 0 by writing a 1 to this bit location.	0	RWC
10:09	DEVT	**DEVSEL# Timing Status:** This 2-bit field defines the timing for DEVSEL# assertion. These read-only bits indicate the DEVSEL# timing when performing a positive decode. Since USB is not PCI-based, the value in this field is arbitrary.	01	RO
08	Reserved	Hardwired as‘0’.	0	
07	Reserved	Hardwired as‘1’	1	
06	Reserved	Hardwired as‘0’.	0	
05	Reserved	Hardwired as‘0’.	0	
04	Reserved	Hardwired as‘0’.	0	
03	INTSTAT	**Interrupt Status:** This read-only bit reflects the state of this function’s interrupt at the input of the enable/disable logic. 0 = Interrupt is deasserted. 1 = Interrupt is asserted. The value reported in this bit is independent of the value in the Interrupt Enable bit.	0	RO
02:00	Reserved	Reserved	000	

25.1.1.4 Offset 08h: Revision ID Register

The value reported in this register depends on the value written to the Revision ID in Device 31, Function 0, Offset 08h.

Table 717. Offset 08h: Revision ID Register

Device: 29	***Function:*** 0, 1
Offset: 08h	***Size:*** 8 bit
Default Value: See Desc	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	REVID	**Revision ID:** 8-bit value that indicates the revision number of the USB1.1 interface.The value reported in this register depends on the value written to the Revision ID in Device 31, Function 0, Offset 08h.	See Desc	RO

25.1.1.5 Offset 09h: Programming Interface Register

Table 718. Offset 09h: Programming Interface Register

Device: 29	***Function:*** 0, 1
Offset: 09h	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	PI	**Programming Interface:** Indicates that no specific register level programming interface defined.	00h	RO

25.1.1.6 Offset 0Ah: Sub Class Code Register

A value of 03h indicates that this is a Universal Serial Bus Host Controller.

Table 719. Offset 0Ah: Sub Class Code Register

Device: 29	***Function:*** 0, 1
Offset: 0Ah	***Size:*** 8 bit
Default Value: 03h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	SCC	**Sub-Class Code:** Indicates the device is a USB1.1 device.	03h	RO

25.1.1.7 Offset 0Bh: Base Class Code Register

A value of 0Ch indicates that this is a Serial Bus controller.

Table 720. Offset 0Bh: Base Class Code Register

Device: 29	***Function:*** 0, 1
Offset: 0Bh	***Size:*** 8 bit
Default Value: 0Ch	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	BCC	**Base Class Code:** Indicates the device is a USB device.	0Ch	RO

25.1.1.8 Offset 0Dh: Master Latency Timer Register

Because the USB controller is internally implemented with arbitration on the Intel® 3100 Chipset, it does not need a master latency timer. The bits are hardwired to 0.

Table 721. Offset 0Dh: Master Latency Timer Register

Device: 29	***Function:*** 0, 1
Offset: 0Dh	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	MLC	**Master Latency Count:** Hardwired to 0.	0h	RO

25.1.1.9 Offset 0Eh: Header Type Register

For Function 1 the register is hard-wired to 00h. For function 0, bit 7 is determined by the values in the USB Function Disable bits (11:8 of the Function Disable register in Memory-mapped configuration space).

Table 722. Offset 0Eh: Header Type Register

Device: 29	***Function:*** 0, 1
Offset: 0Eh	***Size:*** 8 bit
Default Value: FN 0: see below FN 1: 00h FN 2: 00h FN 3: 00h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07	MFB	**Multi-Function Bit:** 0 = Single-function device. 1 = Multi-function device. Since the upper functions in this device can be individually hidden, this bit must be based on the function-disable bits in the Function Disable register in the memory-mapped configuration space as follows: D29_F7_Disable (bit 15) / D29_F3/2/1_Disable (bits 11:9) / Multi-Function Bit 0 / XXX / 1 X / 0XX / 1 X / X0X / 1 X / XX0 / 1 1 / 111 / 0 ***Note:***	See description	RO
06:00	CONFIGL	**Configuration Layout:** Hardwired to 00h, which indicates the standard PCI configuration layout.	00h	RO

25.1.1.10 Offset 20 - 23h: Base Address Register

Table 723. Offset 20 - 23h: Base Address Register

Device: 29	***Function:*** 0, 1
Offset: 20h-23h	***Size:*** 32 bit
Default Value: 00000001h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0	
15:05	BA	**Base Address:** Bits [15:5] correspond to I/O address signals AD [15:5], respectively. This gives 32 bytes of relocatable I/O space.	0	RW
04:01	Reserved	Reserved	0	
00	RTE	**Resource Type Indicator:** This bit is hardwired to 1 indicating that the base address field in this register maps to I/O space	1	RO

25.1.1.11 Offset 2C - 2Dh: USBx_SVID – USB Subsystem Vendor ID Register

Note: X indicates the USB controller number.

BIOS sets the value in this register to identify the Subsystem Vendor ID. The USB_SVID register, in combination with the USB Subsystem ID register, enables the operating system to distinguish each subsystem from the others.

Note: Software can write to this register only once per core well reset. Writes must be done as a single 16-bit cycle.

Table 724. Offset 2C - 2Dh: USBx_SVID – USB Subsystem Vendor ID Register

Device: 29	***Function:*** 0, 1
Offset: 2Ch - 2Dh	***Size:*** 16 bit
Default Value: 0000h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:0	SVID	**Subsystem Vendor ID:** BIOS sets the value in this register to identify the Subsystem Vendor ID. The USB_SVID register, in combination with the USB Subsystem ID register, enables the operating system to distinguish each subsystem from the others.	0000h	RWO

25.1.1.12 Offset 2E - 2Fh: USBx_SID – USB Subsystem ID Register

Note: X indicates the USB controller number.

BIOS sets the value in this register to identify the Subsystem ID. The SID register, in combination with the SVID register, enables the operating system to distinguish each subsystem from other(s).

Note: The software can write to this register only once per core well reset. Writes must be done as a single 16-bit cycle.

Table 725. Offset 2E - 2Fh: USBx_SID – USB Subsystem ID Register

Device: 29	***Function:*** 0, 1
Offset: 2E - 2Fh	***Size:*** 16 bit
Default Value: 0000h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:00	SUBID	**Subsystem ID:** BIOS sets the value in this register to identify the Subsystem ID. The SID register, in combination with the SVID register, enables the operating system to distinguish each subsystem from other(s). Indicates the subsystem as identified by the vendor. This field is write once and is locked until a reset occurs.	0000h	RWO

25.1.1.13 Offset 3Ch: Interrupt Line Register

Table 726. Offset 3Ch: Interrupt Line Register

Device: 29	***Function:*** 0, 1
Offset: 3Ch	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	INTL	**Interrupt line:** This data is not used. This data is used to communicate to software which interrupt line the interrupt pin is connected to.	00h	RW

25.1.1.14 Offset 3Dh: Interrupt Pin Register

Table 727. Offset 3Dh: Interrupt Pin Register

Device: 29	***Function:*** 0, 1
Offset: 3Dh	***Size:*** 8 bit
Default Value: See Description	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	INTPN	**Interrupt Pin:** This read-only value tells the software which interrupt pin each USB host controller uses. The upper 4 bits are hardwired to 0000b; the lower 4 bits are determine by the Interrupt Pin default values that are programmed in the memory-mapped configuration space as follows: Function 0 D29IP.U0P Function 1 D29IP.U1P ***Note:*** This does not determine the mapping to the PIRQ pins.	See description	RO

25.1.1.15 Offset 60h: Serial Bus Release Number Register

A value of 10h indicates that this is controller follows USB release 1.0.

Table 728. Offset 60h: Serial Bus Release Number Register

Device: 29	***Function:*** 0, 1
Offset: 60h	***Size:*** 8 bit
Default Value: 10h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	SBN	Indicates that this controller follows USB release 1.0.	10h	RO

25.1.1.16 Offset C0 - C1h: USB Legacy Keyboard/Mouse Control Register

This register is implemented separately in each of the USB1.1 functions. However, the enable and status bits for the trapping logic are ORed and shared, respectively, since their functionality is not specific to any one host controller.

Table 729. Offset C0 - C1h: USB Legacy Keyboard/Mouse Control Register (Sheet 1 of 3)

Device: 29	***Function:*** 0, 1
Offset: C0h - C1h	***Size:*** 16 bit
Default Value: 2000h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15	SMIBYENDPS	**SMI Caused by End of Pass-through:** Indicates if the event occurred. Even if the corresponding enable bit is not set in the Bit 7, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. 0 = No event occurred. 1 = Event occurred. ***Note:*** Writing a 1 to this bit (in any of the controllers) will clear the latch.	0	RWC
14	Reserved	Reserved	0	
13	USBPIRQEN	**PCI Interrupt Enable:** Used to prevent the USB controller from generating an interrupt due to transactions on its ports. When disabled, that it will probably be configured to generate an SMI using bit 4 of this register. Defaults to 1 for compatibility with older USB software. 0 = Disable 1 = Enable	1	RW
12	SMIBYUSB	**SMI Caused by USB Interrupt:** This bit indicates if an interrupt event occurred from this classic controller. The interrupts from the classic USB controller is taken before the enable in bit 13 has any effect to create this read-only bit. Even if the corresponding enable bit is not set in the Bit 4, then this bit may still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. 0 = No event occurred 1 = Event occurred. ***Note:*** Writing a 1 to this bit will have no effect. The software must clear the interrupts via the USB controllers.	0	RO

Table 729. Offset C0 - C1h: USB Legacy Keyboard / Mouse Control Register (Sheet 2 of 3)

Device: 29	**Function:** 0, 1
Offset: C0h - C1h	**Size:** 16 bit
Default Value: 2000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
11	TRAPBY64W	**SMI Caused by Port 64 Write:** Indicates if the event occurred. Even if the corresponding enable bit is not set in the Bit 3, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit (in any of the controllers) will clear the bit. The A20Gate Pass-Through Logic allows specific port 64h Writes to complete without setting this bit. 0 = No event occurred. 1 = Event Occurred. ***Note:*** Software clears this bit by writing a 1 to the bit location in any of the controllers	0	RWC
10	TRAPBY64R	**SMI Caused by Port 64 Read:** Indicates if the event occurs. Even if the corresponding enable bit is not set in the Bit 2, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit (in any of the controllers) will clear the bit. 0 = No event occurred. 1 = Event Occurred. ***Note:*** Software clears this bit by writing a 1 to the bit location in any of the controllers	0	RWC
09	TRAPBY60W	**SMI Caused by Port 60 Write:** Indicates if the event occurs. Even if the corresponding enable bit is not set in the Bit 1, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit (in any of the controllers) will clear the latch. The A20Gate Pass-Through Logic allows specific port 60h Writes to complete without setting this bit.	0	RWC
08	TRAPBY60R	**SMI Caused by Port 60 Read:** Indicates if the event occurs. Even if the corresponding enable bit is not set in the Bit 0, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit (in any of the controllers) will clear the latch.	0	RWC
07	SMIATENDPS	**SMI at End of Pass-through Enable:** This bit enables SMI at the end of a pass-through. This can occur if an SMI is generated in the middle of a pass through, and needs to be serviced later. 0 = Disable 1 = Enable	0	RW
06	PSTATE	**Pass Through State:** 0 = If software needs to reset this bit, it must set bit 5 in all of the host controllers to 0. 1 = Indicates that the state machine is in the middle of an A20GATE pass-through sequence. ***Note:*** Software must set bit 5 in all of the host controllers to 0 to reset this bit.	0	RO
05	A20PASSEN	**A20Gate Pass-Through Enable:** 0 = Disable. 1 = Enable. Allows A20GATE sequence Pass-Through function. A specific cycle sequence involving writes to port 60h and 64h does not result in the setting of the SMI status bits. SMI# will not be generated, even if the various enable bits are set.	0	RW
04	USBSMIEN	**SMI on USB IRQ:** 0 = Disable. The USB interrupt will not cause an SMI event. 1 = Enable. The USB interrupt will cause an SMI event.	0	RW
03	64WEN	**SMI on Port 64 Writes Enable:** 0 = Disable. A 1 in bit 11 will not cause an SMI event. 1 = Enable. A 1 in bit 11 will cause an SMI event.	0	RW
02	64REN	**SMI on Port 64 Reads Enable:** 0 = Disable. A 1 in bit 10 will not cause an SMI event. 1 = Enable. A 1 in bit 10 will cause an SMI event.	0	RW

Table 729. Offset C0 - C1h: USB Legacy Keyboard/Mouse Control Register (Sheet 3 of 3)

Device: 29	**Function:** 0, 1
Offset: C0h - C1h	**Size:** 16 bit
Default Value: 2000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
01	60WEN	SMI on Port 60 Writes Enable: 0 = Disable. A 1 in bit 9 will not cause an SMI event. 1 = Enable. A 1 in bit 9 will cause an SMI event.	0	RW
00	60REN	**SMI on Port 60 Reads Enable:** 0 = Disable. A 1 in bit 8 will not cause an SMI event. 1 = Enable. A 1 in bit 8 will cause an SMI event.	0	RW

25.1.1.17 Offset C4h: USB Resume Enable Register

Table 730. Offset C4h: USB Resume Enable Register

Device: 29	**Function:** 0, 1
Offset: C4h	**Size:** 8 bit
Default Value: 00h	**Power Well:** Resume

Bits	Name	Description	Reset Value	Access
07:02	Reserved	Reserved	0	
01	PORT1EN	0 = The USB controller will not look at this port for a wakeup event. 1 = Enables port 1 of the USB controller to look at wakeup events. When set, the USB controller will monitor port 1 for remote wakeup and connect/disconnect events. For function 1 this will be port 3	0	RW
00	PORT0EN	0 = The USB controller will not look at this port for a wakeup event. 1 = Enables port 0 of the USB controller to look at wakeup events. When set, the USB controller will monitor port 0 for remote wakeup and connect/disconnect events. For function 1 this will be port 2	0	RW

25.1.1.18 Offset C8h: USB Core Well Policy Register

Table 731. Offset C8h: USB Core Well Policy Register

Device: 29	**Function:** 0, 1
Offset: C8h	**Size:** 8 bit
Default Value: 00h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
07:01	Reserved	Reserved	0	
00	SBMSPEN	**Static Bus Master Status Policy Enable:** 0 = The UHCI Host Controller dynamically sets the Bus Master status bit based on the memory accesses that are scheduled. See Section 25.5 for details. 1 = The UHCI host controller statically forces the Bus Master Status bit in power management space to 1 whenever the HCHalted bit is cleared.	0	RW

25.1.1.19 Offset F8 - FBh: Manufacturer's ID Register

Table 732. Offset F8 - FBh: Manufacturer's ID Register

Device: 29	***Function:*** 0, 1
Offset: F8h-FBh	***Size:*** 32 bit
Default Value: 00010F80h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:24	Reserved	Reserved	00h	
23:16	SID	**Stepping ID:** This field increments for each stepping of the part. This field can be used by software to differentiate steppings when the Revision ID may not change.	01h - A1	RO
15:08	MID	**Manufacturer:** 0Fh = Intel	0Fh	RO
07:00	Reserved	Reserved	80h	

25.2 USB I/O Registers

Some of the read/write register bits which deal with changing the state of the USB hub ports function such that on read back they reflect the current state of the port and not necessarily the state of the last write to the register. This allows the software to poll the state of the port and wait until it is in the proper state before proceeding. A Host controller reset, global reset, or port reset will immediately terminate a transfer on the affected ports and disable the port. This affects the USBCMD register, bit [4] and the PORTSC registers, bits [12,6,2]. See individual bit descriptions for more detail.

Base address for these I/O registers is found in Section 25.1.1.10.

Table 733. USB I/O Registers Summary

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00	01h	USBCMD	USB Command Register	0000h	RW[1]
02	03h	USBSTS	USB Status Register	0020h	RWC
04	05h	USBINTR	USB Interrupt Enable Register	0000h	RW
06	07h	FRNUM	USB Frame Number Register	0000h	RW[1]
08	0Bh	FRBASEADD	USB Frame List Base Address Register	XXXXX000	RW
0C	0Ch	SOFMOD	USB Start of Frame Modify Register	40h	RW
10	13h	PORTSC[0,1]	Port [0,1] Status/Control Register	0080h	RWC[1]

Note:
1. These registers are WORD writable only. Byte writes to these registers have unpredictable effects.
2. The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

25.2.1 Register Details

25.2.1.1 Offset 00 - 01h: USBCMD - USB Command Register

The Command Register indicates the command to be executed by the serial bus host controller. Writing to the register causes a command to be executed. Table 735 provides additional information on the operation of the Run/Stop and Debug bits.

Table 734. Offset 00 - 01h: USBCMD - USB Command Register (Sheet 1 of 2)

I/O Address: 00h - 01h ***Size:*** 16 bit

Default Value: 0000h ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:09	Reserved	Reserved.	0	
08	LBTM	**Loop Back Test Mode:** 0 = Disable. The Intel® 3100 Chipset is not in loop-back test mode. 1 = Enable. The Intel® 3100 Chipset is in loop-back test mode. When both ports are connected together, a write to one port will be seen on the other port and the data will be store in I/O offset 18h.	0	RW
07	MAXP	**Max Packet:** 0 = 32 bytes. 1 = 64 bytes. This bit selects the maximum packet size that can be used for full speed bandwidth reclamation at the end of a frame. This value is used by the Host Controller to determine whether it should initiate another transaction based on the time remaining in the SOF counter. Use of reclamation packets larger than the programmed size will cause a Babble error if executed during the critical window at frame end. The Babble error results in the offending endpoint being stalled. Software is responsible for ensuring that any packet which could be executed under bandwidth reclamation be within this size limit.	0	RW
06	CF	**Configure Flag:** 0 = Indicates that software has not completed host controller configuration. 1 = HCD software sets this bit as the last action in its process of configuring the host controller. This bit has no effect on the hardware. It is provided only as a semaphore service for software.	0	RW
05	SWDBG	**Software Debug:** 0 = Normal Mode. 1 = Debug mode. In software Debug mode, the Host Controller clears the Run/Stop bit after the completion of each USB transaction. The next transaction is executed when software sets the Run/Stop bit back to 1. The SWDBG bit must only be manipulated when the controller is in the stopped state. This can be determined by checking the HCHalted bit in the USBSTS register.	0	RW
04	FGR	**Force Global Resume:** 0 = Software sets this bit to 0 after 20 ms has elapsed to stop sending the Global Resume signal. At that time all USB devices must be ready for bus activity. 1 = Host Controller sends the Global Resume signal on the USB. The Host Controller sets this bit to 1 when a resume event (connect, disconnect, or K-state) is detected while in global suspend mode. The 1 to 0 transition causes the port to send a low speed EOP signal. This bit will remain a 1 until the EOP has completed.	0	RW
03	EGSM	**Enter Global Suspend Mode:** 0 = Software resets this bit to 0 to come out of Global Suspend mode. Software writes this bit to 0 at the same time that Force Global Resume (bit 4) is written to 0 or after writing bit 4 to 0. 1 = Host Controller enters the Global Suspend mode. No USB transactions occurs during this time. The Host Controller is able to receive resume signals from USB and interrupt the system. Software must ensure that the Run/Stop bit (bit 0) is cleared prior to setting this bit.	0	RW

Table 734. Offset 00 - 01h: USBCMD - USB Command Register (Sheet 2 of 2)

I/O Address: 00h - 01h	***Size:*** 16 bit
Default Value: 0000h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
02	GRESET	**Global Reset:** 0 = This bit is reset by the software after a minimum of 10 ms has elapsed as specified in Chapter 7 of the *Universal Serial Bus (USB) Specification, Rev. 2.0*. 1 = Global Reset. The host controller sends the global reset signal on the USB and then resets all its logic, including the NSI registers. The NSI registers are reset to their power on state. Chip Hardware Reset has the same effect as Global Reset (bit 2), except that the host controller does not send the Global Reset on USB.	0	RW
01	HCRESET	**Host Controller Reset:** The HCReset effects on hub registers are slightly different from Chip Hardware Reset and Global USB Reset. The HCReset affects bits [8,3:0] of the Port Status and Control Register (PORTSC) of each port. HCReset resets the state machines of the Host Controller including the Connect/Disconnect state machine (one for each port). When the Connect/Disconnect state machine is reset, the output that signals connect/disconnect are negated to 0, effectively signaling a disconnect, even if a device is attached to the port. This virtual disconnect causes the port to be disabled. This disconnect and disabling of the port causes bit 1 (connect status change) and bit 3 (port enable/disable change) of the PORTSC to get set. The disconnect also causes bit 8 of PORTSC to reset. About 64 bit times after HCReset goes to 0, the connect and low-speed detect will take place and bits 0 and 8 of the PORTSC will change accordingly. 0 = Reset by the host controller when the reset process is complete. 1 = Reset. When this bit is set, the host controller module resets its internal timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated.	0	RW
00	RS	**Run/Stop:** 0 = Stop. Completes the current transaction on the USB and then halts. The HC Halted bit in the status register indicates when the Host Controller has finished the transaction and has entered the stopped state. The Host Controller clears this bit when the following fatal errors occur: consistency check failure, Memory access errors. 1 = Run. Proceeds with execution of the schedule and continues execution as long as this bit is set.	0	RW

Table 735. Run/Stop, Debug Bit Interaction SWDBG (Bit 5), Run/Stop (Bit 0) Operation

SWDBG (Bit 5)	Run/Stop (Bit 0)	Description
0	0	If executing a command, the Host Controller completes the command and then stops. The 1.0 ms frame counter is reset and command list execution resumes from start of frame using the frame list pointer selected by the current value in the FRNUM register. (While Run/Stop=0, the FRNUM register can be reprogrammed).

Table 735. Run/Stop, Debug Bit Interaction SWDBG (Bit 5), Run/Stop (Bit 0) Operation

0	1	Execution of the command list resumes from Start Of Frame using the frame list pointer selected by the current value in the FRNUM register. The Host Controller remains running until the Run/Stop bit is cleared (by Software or Hardware).
1	0	If executing a command, the Host Controller completes the command and then stops and the 1.0 ms frame counter is frozen at its current value. All status are preserved. The Host Controller begins execution of the command list from where it left off when the Run/Stop bit is set.
1	1	Execution of the command list resumes from where the previous execution stopped. The Run/Stop bit is set to 0 by the Host Controller when a TD is being fetched. This causes the Host Controller to stop again after the execution of the TD (single step). When the Host Controller has completed execution, the HC Halted bit in the Status Register is set.

When the USB Host Controller is in Software Debug Mode (USBCMD Register bit 5=1), the single stepping software debug operation is as follows:

To Enter Software Debug Mode:

1. HCD puts Host Controller in Stop state by setting the Run/Stop bit to 0.
2. HCD puts Host Controller in Debug Mode by setting the SWDBG bit to 1.
3. HCD sets up the correct command list and Start Of Frame value for starting point in the Frame List Single Step Loop:
4. HCD sets Run/Stop bit to 1.
5. Host Controller executes next active TD, sets Run/Stop bit to 0, and stops.
6. HCD reads the USBCMD register to check if the single step execution is completed (HCHalted=1).
7. HCD checks results of TD execution. Go to step 4 to execute next TD or step 8 to end Software Debug mode.
8. HCD ends Software Debug mode by setting SWDBG bit to 0.
9. HCD sets up normal command list and Frame List table.
10. HCD sets Run/Stop bit to 1 to resume normal schedule execution.

In Software Debug mode, when the Run/Stop bit is set, the Host Controller starts. When a valid TD is found, the Run/Stop bit is reset. When the TD is finished, the HCHalted bit in the USBSTS register (bit 5) is set.

The software Debug mode skips over inactive TDs and only halts after an active TD has been executed. When the last active TD in a frame has been executed, the Host Controller waits until the next SOF is sent and then fetches the first TD of the next frame before halting.

This HCHalted bit can also be used outside of Software Debug mode to indicate when the Host Controller has detected the Run/Stop bit and has completed the current transaction. Outside of the Software Debug mode, setting the Run/Stop bit to 0 always resets the SOF counter so that when the Run/Stop bit is set the Host Controller starts over again from the frame list location pointed to by the Frame List Index (see FRNUM Register description) rather than continuing where it stopped.

25.2.1.2 Offset 02 - 03h: USBSTS - USB Status Register

This register indicates pending interrupts and various states of the Host Controller. The status resulting from a transaction on the serial bus is not indicated in this register. Software sets a bit to 0 in this register by writing a 1 to it. See Section 4, “Interrupts,” of the *Universal Host Controller Interface (UHCI) Specification, Rev. 1.1*, for additional information concerning USB interrupt conditions.

Table 736. Offset 02 - 03h: USBSTS - USB Status Register

I/O Address: 02h - 03h	***Size:*** 16 bit
Default Value: 0020h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:06	Reserved	Reserved.	0	
05	HCH	**HCHalted:** 0 = The host controller is running. 1 = The host controller has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the host controller hardware (debug mode or an internal error). Default. Software clears this bit by writing a 1 to it.	1	RWC
04	HCPE	**Host Controller Process Error:** 0 = No error. 1 = The host controller has detected a fatal error. This indicates that the Host Controller suffered a consistency check failure while processing a Transfer Descriptor. An example of a consistency check failure would be finding an illegal PID field while processing the packet header portion of the TD. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further schedule execution. A hardware interrupt is generated to the system. Software clears this bit by writing a 1 to it.	0	RWC
03	HSE	**Host System Error:** 0 = No error occurred. 1 = A serious error occurs during a host system access involving the Host Controller module. Conditions that set this bit to 1 include Parity error, Master Abort, and Target Abort. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs. A hardware interrupt is generated to the system. Software clears this bit by writing a 1 to it.	0	RWC
02	RSM_DET	**Resume Detect:** 0 = **Resume not detected.** 1 = The host controller received a “RESUME” signal from a USB device. This is only valid if the Host controller is in a global suspend state (bit 3 of Command register = 1). Software clears this bit by writing a 1 to it.	0	RWC
01	USBEINT	**USB Error Interrupt:** The Host Controller sets this bit to 1 when completion of a USB transaction results in an error condition (e.g., error counter underflow). If the TD on which the error interrupt occurred also had its IOC bit set, both this bit and Bit 0 are set. 0 = No USB Error Interrupt. 1 = Completion of a USB transaction results in an error condition (e.g., error counter underflow). If the TD on which the error interrupt occurred also had its IOC bit set, both this bit and Bit 0 are set. Software clears this bit by writing a 1 to it.	0	RWC
00	USBINT	**USB Interrupt:** 0 = No USB interrupt. 1 = The Host Controller sets this bit to 1 when the cause of an interrupt is a completion of a USB transaction whose Transfer Descriptor had its IOC bit set. The Host Controller also sets this bit to 1 when a short packet is detected (actual length field in TD is less than maximum length field in TD), and short packet detection is enabled in that TD. Software clears this bit by writing a 1 to it.	0	RWC

25.2.1.3 Offset 04 - 05h: USBINTR - USB Interrupt Enable Register

This register enables and disables reporting of the corresponding interrupt to the software. When a bit is set and the corresponding interrupt is active, an interrupt is generated to the host. Fatal errors (Host Controller Processor Error-bit 4, USBSTS Register) cannot be disabled by the host controller. Interrupt sources that are disabled in this register still appear in the Status Register to allow the software to poll for events.

Table 737. Offset 04 - 05h: USBINTR - USB Interrupt Enable Register

I/O Address: 04h - 05h ***Size:*** 16 bit
Default Value: 0000h ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:05	Reserved	Reserved.	0	
04	Scratchpad	Scratchpad.	0	RW
03	SPIEN	**Short Packet Interrupt Enable:** 0 = Disabled 1 = Enabled.	0	RW
02	IOC	**Interrupt On Complete:** 0 = Disabled 1 = Enabled.	0	RW
01	RIEN	**Resume Interrupt Enable:** 0 = Disabled 1 = Enabled.	0	RW
00	TIEN	**Timeout/CRC Interrupt Enable:** 0 = Disabled 1 = Enabled.	0	RW

25.2.1.4 Offset 06 - 07h: FRNUM - Frame Number Register

Bits [10:0] of this register contain the current frame number which is included in the frame SOF packet. This register reflects the count value of the internal frame number counter. Bits [9:0] are used to select a particular entry in the Frame List during schedule execution. This register is updated at the end of each frame time.

This register must be written as a word. Byte writes are not supported. This register cannot be written unless the Host Controller is in the STOPPED state as indicated by the HCHalted bit (USBSTS register). A write to this register while the Run/Stop bit is set (USBCMD register) is ignored.

Table 738. Offset 06 - 07h: FRNUM - Frame Number Register

I/O Address: 06h - 07h ***Size:*** 16 bit
Default Value: 0000h ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:11	Reserved	Reserved.	0	
10:00	FLCIFN	**Frame List Current Index/Frame Number:** Bits [10:0] provide the frame number in the SOF Frame. The value in this register increments at the end of each time frame (approximately every 1 ms). In addition, bits [9:0] are used for the Frame List current index and correspond to memory address signals [11:2].	0	RW

25.2.1.5 Offset 08 - 0Bh: FRBASEADD - Frame List Base Address Register

This 32-bit register contains the beginning address of the Frame List in the system memory. HCD loads this register prior to starting the schedule execution by the Host Controller. When written, only the upper 20 bits are used. The lower 12 bits are written as zero (4 Kbyte alignment). The contents of this register are combined with the frame number counter to enable the Host Controller to step through the Frame List in sequence. The two least significant bits are always 00. This requires DWord alignment for all list entries. This configuration supports 1024 Frame List entries.

Table 739. Offset 08 - 0Bh: FRBASEADD - Frame List Base Address Register

I/O Address: 08h - 0Bh ***Size:*** 32 bit
Default Value: XXXXX000 ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:12	BAD	**Base Address:** These bits correspond to memory address signals [31:12], respectively.	X	R/W
11:00	Reserved	Reserved. Must be written as 0s.	0	

25.2.1.6 Offset 0Ch: SOFMOD - Start of Frame Modify Register

This 1-byte register is used to modify the value used in the generation of SOF timing on the USB. Only the 7 least significant bits are used. When a new value is written into the these 7 bits, the SOF timing of the next frame will be adjusted. This feature can be used to adjust out any offset from the clock source that generates the clock that drives the SOF counter. This register can also be used to maintain real time synchronization with the rest of the system so that all devices have the same sense of real time. Using this register, the frame length can be adjusted across the full range required by the *USB Specification*. Its initial programmed value is system dependent based on the accuracy of hardware USB clock and is initialized by system BIOS. It may be reprogrammed by USB system software at any time. Its value will take effect from the beginning of the next frame. This register is reset upon a Host Controller Reset or Global Reset. Software must maintain a copy of its value for reprogramming if necessary.

Table 740. Offset 0Ch: SOFMOD - Start of Frame Modify Register (Sheet 1 of 2)

I/O Address: 0Ch ***Size:*** 8 bit
Default Value: 40h ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07	Reserved	Reserved.	0	

Table 740. Offset 0Ch: SOFMOD - Start of Frame Modify Register (Sheet 2 of 2)

I/O Address: 0Ch		*Size:* 8 bit		
Default Value: 40h		*Power Well:* Core		
Bits	**Name**	**Description**	**Reset Value**	**Access**
06:00	SOFTV	**SOF Timing Value:** Guidelines for the modification of frame time are contained in Chapter 7 of the *USB Specification*. The SOF cycle time (number of SOF counter clock periods to generate a SOF frame length) is equal to 11936 + value in this field. The default value is decimal 64 which gives a SOF cycle time of 12000. For a 12 MHz SOF counter clock input, this produces a 1 ms Frame period. The following table indicates what SOF Timing Value to program into this field for a certain frame period. Frame Length (# 12 MHz Clocks) (decimal) / SOF Timing Value (this register) (decimal) 11936 / 0 11937 / 1 — / — 11999 / 63 12000 / 64 12001 / 65 — / — 12062 / 126	1000000	R/W

25.2.1.7 Offset 10h, 12h: Port Status and Control Register

After a Power-up reset, Global reset, or Host Controller reset, the initial conditions of a port are: No device connected, Port disabled, and the bus line status is 00 (single-ended zero).

Table 741. Offset 10h, 12h: Port Status and Control Register (Sheet 1 of 3)

I/O Address: USB #1 Port 0: 10-11h, Port 1: 12-13h USB #2 Port 2: 10-11h, Port 3: 12-13h		*Size:* 16 bit		
Default Value: 0080h		*Power Well:* Core		
Bits	**Name**	**Description**	**Reset Value**	**Access**
15:13	Reserved	Reserved	0	

Table 741. Offset 10h, 12h: Port Status and Control Register (Sheet 2 of 3)

I/O Address: USB #1 Port 0: 10-11h, Port 1: 12-13h USB #2 Port 2: 10-11h, Port 3: 12-13h			***Size:*** 16 bit	
Default Value: 0080h			***Power Well:*** Core	
Bits	**Name**	**Description**	**Reset Value**	**Access**
12	SUS	**Suspend:** 0 = Port not in suspend state. 1 = Port in suspend state. This bit should not be written to a 1 if global suspend is active (bit 3=1 in the USBCMD register). This bit and bit 2 below define the hub states: Bits [12,2] Hub State X,0 Disable 0,1 Enable 1,1 Suspend When in suspend state, downstream propagation of data is blocked on this port, except for single-ended 0 resets (global reset and port reset). The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. The bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB. Normally, if a transaction is in progress when this bit is set, the port will be suspended when the current transaction completes. However, in the case of a specific error condition (out transaction with babble), Intel® 3100 Chipset may issue a start-of-frame, and then suspend the port.	0	RW
11	OCI	**Overcurrent Indicator:** This bit is cleared by software writing a '1'. 0 = Overcurrent pin inactive. 1 = Overcurrent pin has gone from inactive to active on this port. Software clears this bit by writing a 1 to it.	0	RWC
10	OCS	**Overcurrent Status:** This bit indicates the current status of the OC# pin for this port. This bit is set and cleared by hardware. 0 = This port does not have an overcurrent condition. 1 = This port currently has an overcurrent condition.	0	RO
09	PRST	**Port Reset:** 0 = Port is not disabled. 1 = Port is disabled and sends the USB Reset signaling.	0	RW
08	LS	**Low Speed Device Attached:** 0 = Full speed device is attached. 1 = Low speed device is attached to this port. Writes have no effect.	0	RO
07	Reserved	Reserved: Always read as 1.	1	
06	RSM_DET	**Resume Detect:** 0 = No resume (K-state) detected/driven on port. 1 = Resume detected/driven on port. Software sets this bit to a 1 to drive resume signaling. The Host Controller sets this bit to a 1 if a J-to-K transition is detected for at least 32 µs while the port is in the Suspend state. The Intel® 3100 Chipset will then reflect the K-state back onto the bus as long as the bit remains a '1', and the port is still in the suspend state (bit 12,2 are '11') Writing a 0 (from 1) causes the port to send a low speed EOP. This bit will remain a 1 until the EOP has completed.	0	RW
05:04	LNS	**Line Status:** These bits reflect the D+ (bit 4) and D- (bit 5) signals lines logical levels. These bits are used for fault detect and recovery as well as for USB diagnostics. This field is updated at EOF2 time.	00	RO

Table 741. Offset 10h, 12h: Port Status and Control Register (Sheet 3 of 3)

I/O Address: USB #1 Port 0: 10-11h, Port 1: 12-13h USB #2 Port 2: 10-11h, Port 3: 12-13h			**Size:** 16 bit	
Default Value: 0080h			**Power Well:** Core	
Bits	**Name**	**Description**	**Reset Value**	**Access**
03	PORT_ENC	**Port Enable/Disable Change:** 0 = No change. 1 = Port enabled/disabled status has changed. For the root hub, this bit gets set only when a port is disabled due to disconnect on the that port or due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the *USB Specification*). Software clears this bit by writing a 1 to it.	0	R/WC
02	PORT_EN	**Port Enabled/Disabled:** 0 = Disable. 1 = Enable. Ports can be enabled by host software only. Ports can be disabled by either a fault condition (disconnect event, overcurrent, or other fault condition) or by host software. The bit status does not change until the port state actually changes and that there may be a delay in disabling or enabling a port if there is a transaction currently in progress on the USB.	0	RW
01	CSC	**Connect Status Change:** 0 = No change. 1 = Change in Current Connect Status. Indicates a change has occurred in the port's Current Connect Status (see bit 0). The hub device sets this bit for any changes to the port device connect status, even if system software has not cleared a connect status change. If, for example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be "setting" an already-set bit (i.e., the bit will remain set). However, the hub transfers the change bit only once when the Host Controller requests a data transfer to the Status Change endpoint. System software is responsible for determining state change history in such a case. Software sets this bit to 0 by writing a 1 to it.	0	R/WC
00	CCS	**Current Connect Status:** 0 = No device is present. 1 = Device is present on port. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set.	0	RO

25.3 Data Transfers to/from Main Memory

Section 3.4 of the *Universal Host Controller Interface (UHCI) Specification, Rev. 1.1*describes the details on how HCD and the Intel® 3100 Chipset communicate via the Schedule data structures.

25.4 Data Structures in Main memory

Sections 3.1 through 3.3 of the *UHCI Specification* detail the data structures used to communicate control, status, and data between software and the Intel® 3100 Chipset.

25.5 Data Transfers To/From Main Memory

The following sections describe the details on how HCD and the Intel® 3100 Chipset communicate via the Schedule data structures. The discussion is organized in a top-down manner, beginning with the basics of walking the Frame List, followed by a description of generic processing steps common to all transfer descriptors, and finally a discussion on Transfer Queuing.

During data transfers to and from main memory, the UHCI DMA engine must provide an indication to the processor power management logic that it is busy. Therefore, the memory accesses may actually be “cache accesses”. The indication to the power management logic will be referred to as “UHCI Bus Master Status”.

The UHCI controllers offer two different policies for the generation of the UHCI Bus Master Status: static and dynamic.

The static policy requires that the UHCI Bus Master Status signal is asserted when the HCHalted bit is 0.

The dynamic policy requires that the UHCI Bus Master Status signal deasserts after completely traversing the schedule for a particular Frame unless the Frame bit counter indicates that the next Frame begins in less than ~100 usecs. Specifically the UHCI Bus Master Status signal must deassert in any of the following:

1. After reading the Frame List Pointer and detecting that the Terminate bit is set and the next Frame is greater than ~100 usecs in the future.
2. After servicing a Queue Head in which the Terminate bit is set in the Queue Head Link Pointer and the next Frame is greater than ~100 usecs in the future.
3. The RUN bit is 0 and the HCHalted bit is 1
4. After servicing a Transfer Descriptor which is not in the context of a Queue and it's Terminate bit is set and the next Frame is greater than ~100 µs in the future

The UHCI Bus Master Status signal must assert in either of the following cases:

1. The RUN bit transitions from 0 to 1
2. The Frame bit counter indicates that the next Frame begins in less than ~100 µs

The value of 100 µs is somewhat arbitrary. This is based on the assumption that a “reasonable worst case” exit time is approximately 100 µs. Even if the exit latency is greater than this, it does not mean that there will be a USB functional failure. It only means that the USB traffic will begin later in the upcoming Frame than it normally would have. In order to specify numbers for checking in validation, the ~100 µs time must fall within the range 97 µs to 110 µs.

The UHCI BM_STS Static Policy Enable configuration bit (D31.F0.A9h.5) selects between the dynamic and static policies.

25.5.1 Executing the Schedule

Software programs the starting address of the Frame List and the Frame List index, then causes the Intel® 3100 Chipset to execute the schedule by setting the Run/Stop bit in the Control register to Run. The Intel® 3100 Chipset processes the schedule one entry at a time: the next element in the frame list is not fetched until the current element in the frame list is retired.

Schedule execution proceeds in the following fashion:

- The Intel® 3100 Chipset first fetches an entry from the Frame List. This entry has three fields. Bit 0 indicates whether the address pointer field is valid. Bit 1 indicates whether the address points to a Transfer Descriptor or to a queue head. The third field is the pointer itself.
- If the Frame List entry indicates that it points to a Transfer Descriptor, the Intel® 3100 Chipset fetches the entry and begins the operations necessary to initiate a transaction on USB. Each TD contains a link field that points to the next entry, as well as indicating whether it is a TD or a QH.
- If the Frame List entry contains a pointer to a QH, the Intel® 3100 Chipset processes the information from the QH to determine the address of the next data object that it must process.
- The TD/QH process continues until the millisecond allotted to the current frame expires. At this point, the Intel® 3100 Chipset fetches the next entry from the Frame List. If the Intel® 3100 Chipset is not able to process all of the transfer descriptors during a given frame, those descriptors are retired by software without having been executed.

25.5.2 Processing Transfer Descriptors

The Intel® 3100 Chipset executes a TD using the following, generalized algorithm. These basic steps are common across all modes of TDs. Subsequent sections present processing steps unique to each TD mode.

1. The Intel® 3100 Chipset Fetches TD or QH from the current Link Pointer.
2. If a QH, go to 1 to fetch from the Queue Element Link Pointer. If inactive, go to 12
3. Build Token, actual bits are in TD Token.
4. If (Host-to-Function) then
   [*Memory Access*] issue request for data, (referenced through TD.BufferPointer)
   wait for first chunk data arrival
   end if
5. [*Begin USB Transaction*] Issue Token (from token built in 2, above) and begin data transfer.
   if (Host-to-Function) then Go to 6
   else Go to 7
   end if
6. Fetch data from memory (via TD BufferPointer) and transfer over USB until TD Maximum-Length bytes have been read and transferred. [*Concurrent system memory and USB Accesses*]. Go to 8.
7. Wait for data to arrive (from USB). Write incoming bytes into memory beginning at TD BufferPointer. Internal HC buffer must signal end of data packet. Number of bytes received must be ≤ TD Maximum-Length; The length of the memory area referenced by TD BufferPointer must be ≥ TD Maximum-Length. [*Concurrent system memory and USB Accesses*].
8. Issue handshake based on status of data received (Ack or Time-out). Go to 10.
9. Wait for handshake, if required [*End of USB Transaction*].

10. Update Status [*Memory Access*] (TD.Status and TD.ActualLength).
    If the TD completed successfully, mark the TD inactive. Go to 11.
    If not successful, and the error count has not been reached, leave the TD active. If the error count has been reached, mark the TD inactive. Go to 12.
11. Write the link pointer from the current TD into the element pointer field of the QH structure. If the Vf bit is set in the link pointer, go to 2.
12. Proceed to next entry.

25.5.3 Command Register, Status Register, and TD Status Bit Interaction

Condition	USB Status Register Actions	TD Status Register Actions
CRC/Time Out Error	Set USB Error Int bit[1], Clear HC Halted bit	Clear Active bit[1] and set Stall bit[1]
Illegal PID, PID Error, Maximum Length (illegal)	Clear Run/Stop bit in command register Set HC Process Error and HC Halted bits	
Master/Target Abort on memory accesses	Clear Run/Stop bit in command register Set Host System Error and HC Halted bits	
Suspend Mode	Clear Run/Stop bit in command register[2] Set HC Halted bit	
Resume Received Suspend Mode = 1	Set Resume received bit	
Run/Stop = 0	Clear Run/Stop bit in command register Set HC Halted bit	
Configuration Flag Set	Set Configuration Flag in command register	
HC Reset/Global Reset	Clear Run/Stop and configuration Flag in command register Clear USB Int, USB Error Int, Resume received, Host System Error, HC Process Error, and HC Halted bits	
IOC = 1 in TD Status	Set USB Int bit	
Stall	Set USB Error Int bit	Clear Active bit[1] and set Stall bit
Bit Stuff/Data Buffer Error	Set USB Error Int bit[1]	Clear Active bit[1] and set Stall bit[1]
Short Packet Detect	Set USB Int bit	Clear Active bit

Notes:
1. Only If error counter counted down from 1 to 0.
2. Suspend mode can be entered only when Run/Stop bit is 0.

Note: If a NAK or STALL response is received from a SETUP transaction, a Time Out Error will be reported. This will cause the Error counter to decrement and the CRC/Time-out Error status bit to be set within the TD Control and Status Dword during write back. If the Error counter changes from 1 to 0, the Active bit will be reset to 0 and Stalled bit to 1 as normal.

25.5.4 Transfer Queuing

Transfer Queues are used to implement a guaranteed data delivery stream to a USB Endpoint. Transfer Queues are composed of two parts: a Queue Header (QH) and a linked list. The linked list of TDs and QHs has an indeterminate length (0 to n).

The QH contains two link pointers and is organized as two contiguous DWords. The first DWord is a horizontal pointer (Queue Head Link Pointer), used to link a single transfer queue with either another transfer queue, or a TD (target data structure depends on Q bit). If the T bit is set, this QH represents the last data structure in the current Frame. The T bit informs the Intel® 3100 Chipset that no further processing is required until

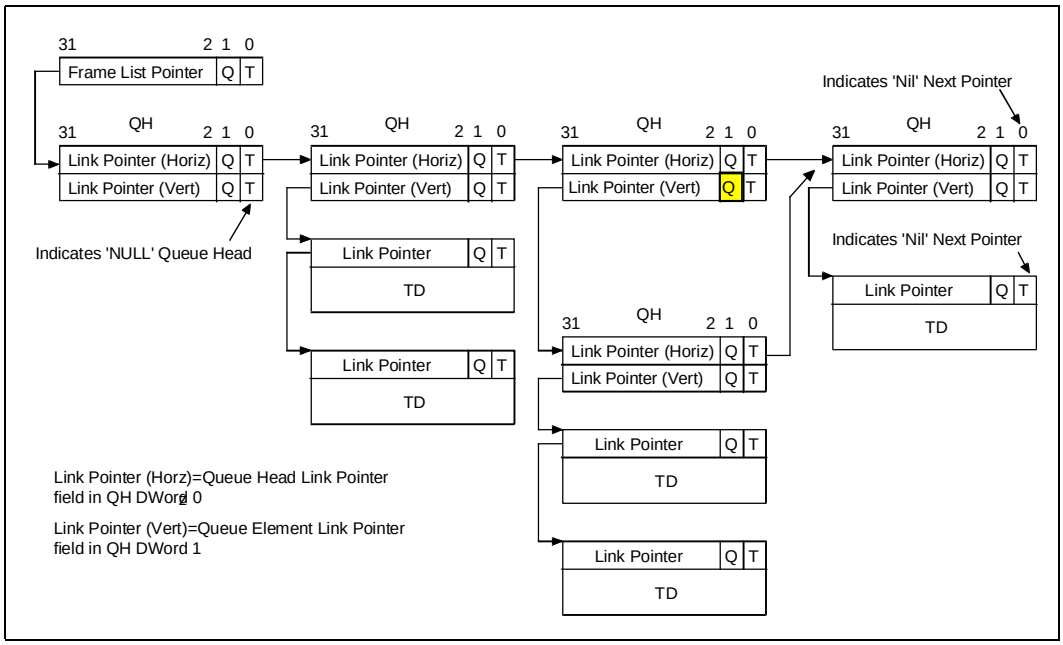
the beginning of the next frame. The second DWord is a vertical pointer (Queue Element Link Pointer) to the first data structure (TD or QH) being managed by this QH. If the T bit is set, the queue is empty. This pointer may reference a TD or another QH.

Figure 83 illustrates four example queue conditions. The first QH (on far left) is an example of an “empty” queue; the termination bit (T Bit), in the vertical link pointer field, is set to 1. The horizontal link pointer references another QH. The next queue is the expected typical configuration. The horizontal link pointer references another QH, and the vertical link pointer references a valid TD.

Typically, the vertical pointer in a QH points to a TD. However, as shown in Figure 83 (third example from left side of figure) the vertical pointer could point to another QH. When this occurs, a new Q Context is entered and the Q Context just exited is NULL (The Intel® 3100 Chipset will not update the vertical pointer field).

The far right QH is an example of a frame ‘termination’ node. Since its horizontal link pointer has its termination bit set, the Intel® 3100 Chipset assumes there is no more work to complete for the current Frame.

Figure 83. Example Queue Conditions



Transfer Queues are based on the following characteristics:

- A QH’s vertical link pointer (Queue Element Link Pointer) references the ‘Top’ queue member. A QH’s horizontal link pointer (Queue Head Link Pointer) references the “next” work element in the Frame.
- Each queue member’s link pointer references the next element within the queue.

In the simplest model, the Intel® 3100 Chipset follows vertical link point to a queue element, then executes the element. If the completion status of the TD satisfies the advance criteria as shown in Table 742, the Intel® 3100 Chipset advances the queue by writing the just-executed TD’s link pointer back into the QH’s Queue Element link pointer. The next time the queue head is traversed, the next queue element will be the Top element.

The traversal has two options: Breadth first, or Depth first. A flag bit in each TD (Vf - Vertical Traversal Flag) controls whether traversal is Breadth or Depth first. The default mode of traversal is Breadth-First. For Breadth-First, the Intel® 3100 Chipset only executes the top element from each queue. The execution path is:

QH (Queue Element Link Pointer)→ TD→ Write-Back to QH (Queue Element Link Pointer)→ QH (Queue Head Link pointer).

Breadth-First is also performed for every transaction execution that fails the advance criteria. This means that if a queued TD fails, the queue does not advance, and the Intel® 3100 Chipset traverses the QH's Queue Head Link Pointer.

In a Depth-first traversal, the top queue element must complete successfully to satisfy the *advance criteria* for the queue. If the Intel® 3100 Chipset is currently processing a queue, and the advance criteria are met, and the Vf bit is set, the Intel® 3100 Chipset follows the TD's link pointer to the next schedule work item.

Regardless of traversal model, when the advance criteria are met, the successful TD's link pointer is written back to the QH's Queue Element link pointer.

When the Intel® 3100 Chipset encounters a QH, it caches the QH internally, and sets internal state to indicate it is in a Q-context. It needs this state to update the correct QH (for auto advancement) and also to make the correct decisions on how to traverse the Frame List.

Restricting the advancement of queues to advancement criteria implements a guaranteed data delivery stream.

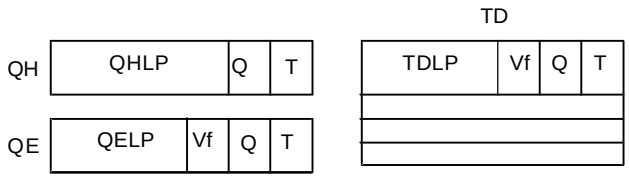
A queue is **never** advanced on an error completion status (even in the event the error count was exhausted).

Table 742 lists the general queue advance criteria, which are based on the execution status of the TD at the 'Top' of a currently 'active' queue.

Table 742. Queue Advance Criteria

Function-to-Host (IN)			Host-to-Function (OUT)		
Non-NULL	**NULL**	**Error/NAK**	**Non-NULL**	**NULL**	**Error/NAK**
Advance Q	Advance Q	Retry Q Element	Advance Q	Advance Q	Retry Q Element

Table 743 is a decision table illustrating the valid combinations of link pointer bits and the valid actions taken when advancement criteria for a queued transfer descriptor are met. The column headings for the link pointer fields are encoded, based on the following list:



Legend:
QH.LP = Queue Head Link Pointer (or Horizontal Link Pointer)QE.Q = Q bit in QE
QE.LP = Queue Element Link Pointer (or Vertical Link Pointer)QE.T = T bit in QE
TD.LP = TD Link PointerTD. Vf = Vf bit in TD
QH.Q = Q bit in QH TD.Q = Q bit in TD
QH.T = T bit in QH TD. Vf = Vf bit in TD

Table 743. USB Schedule List Traversal Decision Table (Sheet 1 of 2)

Q Context	QH.Q	QH.T	QE.Q	QE.T	TD.Vf	TD.Q	TD.T	Description
0	-	-	-	-	X	0	0	Not in Queue – execute TD. Use TD.LP to get next TD
0	-	-	-	-	X	X	1	Not in Queue – execute TD. End of Frame
0	-	-	-	-	X	1	0	Not in Queue - execute TD. Use TD.LP to get next (QH+QE). Set Q Context to 1.
1	0	0	0	0	0	X	X	In Queue. Use QE.LP to get TD. execute TD. Update QE.LP with TD.LP. Use QH.LP to get next TD.
1	X	X	0	0	1	0	0	In Queue. Use QE.LP to get TD. execute TD. Update QE.LP with TD.LP. Use TD.LP to get next TD.
1	X	X	0	0	1	1	0	In Queue. Use QE.LP to get TD. execute TD. Update QE.LP with TD.LP. Use TD.LP to get next (QH+QE).
1	0	0	X	1	X	X	X	In Queue. Empty queue. Use QH.LP to get next TD
1	X	X	1	0	-	-	-	In Queue. Use QE.LP to get (QH+QE)
1	X	1	0	0	0	X	X	In Queue. Use QE.LP to get TD. execute TD. Update QE.LP with TD.LP. End of Frame

Table 743. USB Schedule List Traversal Decision Table (Sheet 2 of 2)

Q Context	QH.Q	QH.T	QE.Q	QE.T	TD.Vf	TD.Q	TD.T	Description
1	X	1	X	1	X	X	X	In Queue. Empty queue. End of Frame
1	1	0	0	0	0	X	X	In Queue. Use QE.LP to get TD. execute TD. Update QE.LP with TD.LP. Use QH.LP to get next (QH+QE).
1	1	0	X	1	X	X	X	In Queue. Empty queue. Use QH.LP to get next (QH+QE)

25.6 USB Buffer Management

The USB controller contains a 64 byte FIFO, which operate in a ping/pong fashion. This buffer is not aligned to any memory boundary. Upon each new transfer descriptor (TD), the buffer resets its pointer to the top.

When transmitting to a USB port, if the length field from the TD is greater than 32 bytes, meaning the internal FIFO boundary will be crossed, the host controller will read both ping and pong together. When ping has been transferred to USB, and the total TD length has not been fetched, the USB host controller will fetch another 32 bytes.

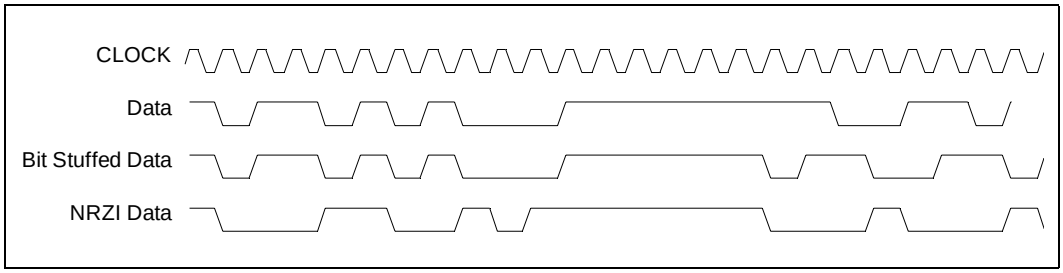
When receiving from a USB port, it is not known how many bytes will be received. Therefore, the Intel® 3100 Chipset will flush each buffer to memory as it becomes full.

25.7 Data Encoding and Bit Stuffing

The Intel® 3100 Chipset USB employs NRZI data encoding (Non-Return to Zero Inverted) when transmitting packets. Full details on this implementation are given in the *USB Specification, Rev. 2.0*.

The USB employs NRZI data encoding when transmitting packets. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. A string of zeros causes the NRZI data to toggle each bit time. A string of ones causes long periods with no transitions in the data. In order to ensure adequate signal transitions, bit stuffing is employed by the transmitting device when sending a packet on the USB. A 0 is inserted after every six consecutive 1s in the data stream before the data is NRZI encoded to force a transition in the NRZI data stream. This gives the receiver logic a data transition at least once every seven bit times to guarantee the data and clock lock. A waveform of the data encoding is shown in Figure 84.

Figure 84. USB Data Encoding



Bit stuffing is enabled beginning with the Sync Pattern and throughout the entire transmission. The data “one” that ends the Sync Pattern is counted as the first one in a sequence. Bit stuffing is always enforced, without exception. If required by the bit stuffing rules, a zero bit will be inserted even if it is the last bit before the end-of-packet (EOP) signal.

25.8 Bus Protocol

25.8.1 Bit Ordering

Bits are sent out onto the bus least significant bit (LSb) first, followed by next LSB, the most significant bit (MSb) last.

25.8.2 SYNC Field

All packets begin with a synchronization (SYNC) field, which is a coded sequence that generates a maximum edge transition density. The SYNC field appears on the bus as IDLE followed by the binary string “KJKJKJKK,” in its NRZI encoding. It is used by the input circuitry to align incoming data with the local clock and is defined to be eight bits in length. SYNC serves only as a synchronization mechanism and is not shown in the following packet diagrams in Section 25.8.3. The last two bits in the SYNC field are a marker that is used to identify the first bit of the PID. All subsequent bits in the packet must be indexed from this point.

25.8.3 Packet Field Formats

All packets have distinct start and end of packet delimiters. Full details are given in the *USB Specification*.

25.8.4 Address Fields

Function endpoints are addressed using two fields: the function address field and the endpoint field. Full details on this are given in the *USB Specification* in Section 8.3.2.

25.8.5 Frame Number Field

The frame number field is an 11-bit field that increments by the host on a per frame basis. The frame number field rolls over upon reaching its maximum value of x7FF, and is sent only for SOF tokens at the start of each frame.

25.8.6 Data Field

The data field may range from 0 to 1023 bytes and must be an integral numbers of bytes. Table 744 lists the format for multiple bytes. Data bits within each byte are shifted out LSB first.

Table 744. Data Field

Bit	Data Sent
X - 1	Byte N - 1, D7
X	Byte N, D0
X + 1	Byte N, D1
X + 2	Byte N, D2
X + 3	Byte N, D3

Table 744. Data Field

X + 4	Byte N, D4
X + 5	Byte N, D5
X + 6	Byte N, D6
X + 7	Byte N, D7
X + 8	Byte N+1, D 0

25.8.7 Cyclic Redundancy Check (CRC)

CRC is used to protect the all non-PID fields in token and data packets. In this context, these fields are considered to be protected fields. Full details on this are given in the *USB Specification* in Section 8.3.5.

25.9 Packet Formats

The USB protocol calls out several packet types: token, data, and handshake packets. Full details on this are given in the *USB Specification* in Section 8.4.

25.10 .USB Interrupts

25.10.1 Overview

There are two general groups of USB interrupt sources, those resulting from execution of transactions in the schedule, and those resulting from an Intel® 3100 Chipset operation error. All transaction-based sources can be masked by software through the Interrupt Enable register. Additionally, individual transfer descriptors can be marked to generate an interrupt on completion.

When the Intel® 3100 Chipset drives an interrupt for USB, it internally drives the PIRQA# pin for USB function #0and PIRQD# pin for USB function #1 (see Chapter 28.0, "Device 29, Function 7: USB 2.0 Host Controller") until all sources of the interrupt are cleared. In order to accommodate some operating systems, the Interrupt Pin register must contain a different value for each function of this multi-function device.

25.10.2 Transaction-Based Interrupts

These interrupts are not signaled until after the status for the last complete transaction in the frame has been written back to host memory. This guarantees that software can safely process through (Frame List Current Index -1) when it is servicing an interrupt.

25.10.2.1 CRC Error/Time-out

A CRC/Time-out error occurs when a packet transmitted from the Intel® 3100 Chipset to a USB device or a packet transmitted from a USB device to the Intel® 3100 Chipset generates a CRC error. The Intel® 3100 Chipset is informed of this event by a time out from the USB device or by the Intel® 3100 Chipset's CRC checker generating an error on reception of the packet. Additionally, a USB bus time-out occurs when USB devices do not respond to a transaction phase within 19 bit times of an EOP. Either of these conditions will cause the C_ERR field of the TD to be decremented.

When the C_ERR field decrements to zero, the following occurs:

- The Active bit in the TD is cleared
- The Stalled bit in the TD is set

- The CRC/Time-out bit in the TD is set.
- At the end of the frame, the USB Error Interrupt bit is set in the HC status register.

If the CRC/Time out interrupt is enabled in the Interrupt Enable register, a hardware interrupt will be signaled to the system.

25.10.2.2 Interrupt on Completion

Transfer Descriptors contain a bit that can be set to cause an interrupt on their completion. The completion of the transaction associated with that block causes the USB Interrupt bit in the HC Status Register to be set at the end of the frame in which the transfer completed. When a TD is encountered with the IOC bit set to 1, the IOC bit in the HC Status register is set to 1 at the end of the frame if the active bit in the TD is set to 0 (even if it was set to 0 when initially read).

If the IOC Enable bit of Interrupt Enable register (bit 2 of I/O offset 04h) is set, a hardware interrupt is signaled to the system. This status bit is set whether the TD completes successfully, or because of errors. If the completion is because of errors, the USB Error bit in the HC status register is also set.

25.10.2.3 Short Packet Detect

A transfer set is a collection of data which requires more than 1 USB transaction to completely move the data across the USB. An example might be a large print file which requires numerous TDs in multiple frames to completely transfer the data. Reception of a data packet that is less than the endpoint's Maximum Packet size during Control, Bulk or Interrupt transfers signals the completion of the transfer set, even if there are active TDs remaining for this transfer set. Setting the SPD bit in a TD indicates to the HC to set the USB Interrupt bit in the HC status register at the end of the frame in which this event occurs. This feature streamlines the processing of input on these transfer types. If the Short Packet Interrupt Enable bit in the Interrupt Enable register is set, a hardware interrupt is signaled to the system at the end of the frame where the event occurred.

25.10.2.4 Serial Bus Babble

When a device transmits on the USB for a time greater than its assigned Maximum Length, it is said to be babbling. This error results in the Active bit in the TD being cleared to 0 and the Stalled and Babble bits being set to one. The C_ERR field is not decremented for a babble. The USB Error Interrupt bit in the HC Status register is set to 1 at the end of the frame. A hardware interrupt is signaled to the system.

If an EOF babble was caused by the Intel® 3100 Chipset (due to incorrect schedule for instance), the Intel® 3100 Chipset will force a bit stuff error followed by an EOP and the start of the next frame.

25.10.2.5 Stalled

This event indicates that a device/endpoint returned a STALL handshake during a transaction or that the transaction ended in an error condition. The TDs Stalled bit is set and the Active bit is cleared. Reception of a STALL does not decrement the error counter. A hardware interrupt is signaled to the system.

25.10.2.6 Data Buffer Error

This event indicates that an overrun of incoming data or a under-run of outgoing data has occurred for this transaction. This would generally be caused by the Intel® 3100 Chipset not being able to access required data buffers in memory within necessary latency requirements. Either of these conditions will cause the C_ERR field of the TD to be decremented.

When C_ERR decrements to zero, the Active bit in the TD is cleared, the Stalled bit is set, the USB Error Interrupt bit in the HC Status register is set to 1 at the end of the frame and a hardware interrupt is signaled to the system.

25.10.2.7 Bit Stuff Error

A bit stuff error results from the detection of a sequence of more that six 1s in a row within the incoming data stream. This will cause the C_ERR field of the TD to be decremented. When the C_ERR field decrements to 0, the Active bit in the TD is cleared to 0, the Stalled bit is set to 1, the USB Error Interrupt bit in the HC Status register is set to 1 at the end of the frame and a hardware interrupt is signaled to the system.

25.10.3 Non-Transaction Based Interrupts

If a Intel® 3100 Chipset process error or system error occur, the Intel® 3100 Chipset halts and immediately issues a hardware interrupt to the system.

25.10.3.1 Resume Received

This event indicates that the Intel® 3100 Chipset received a RESUME signal from a device on the USB bus during a global suspend. If this interrupt is enabled in the Interrupt Enable register, a hardware interrupt will be signaled to the system allowing the USB to be brought out of the suspend state and returned to normal operation.

25.10.3.2 Process Error

The HC monitors certain critical fields during operation to ensure that it does not process corrupted data structures. These include checking for a valid PID and verifying that the MaxLength field is less than 1280. If it detects a condition that would indicate that it is processing corrupted data structures, it immediately halts processing, sets the HC Process Error bit in the HC Status register and signals a hardware interrupt to the system.

This interrupt cannot be disabled through the Interrupt Enable register.

25.10.3.3 Host System Error

The Intel® 3100 Chipset sets this bit to 1 when a Parity Error, Master Abort, or Target Abort occurs on memory accesses. When this error occurs, the Intel® 3100 Chipset clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs. This interrupt cannot be disabled through the Interrupt Enable register.

25.10.3.4 Implementation Notes

1. If a bad PID is found, but the packet will not be run because there is not enough time left in the frame due to the size of the transfer or the packet time out, an error will not be flagged.
2. “If a bad PID is found, but the packet will not be run because the TD’s active bit is off, the host controller will not halt, and an error will not be flagged.”

25.11 USB Power Management

The Host Controller can be put into a suspended state and its power can be removed. This requires that certain bits of information are retained in the resume power plane of the Intel® 3100 Chipset so that a device on a port may wake the system. Such a device may be a fax-modem, which will wake up the machine to receive a fax or take a voice message.

The following bits in I/O space are to be maintained when the Intel® 3100 Chipset enters a low power state:

Table 745. Bits Maintained in Low Power States

Register	Offset	Bit	Description
Command	00h	3	Enter Global Suspend Mode (EGSM)
Status	02h	2	Resume Detect
Port Status and Control	10h and 12h	2	Port Enabled/Disabled
		6	Resume Detect
		8	Low-speed Device Attached
		12	Suspend

When the Intel® 3100 Chipset detects a resume event on any of its ports, it sets the corresponding USB_STS bit in ACPI space. If USB is enabled as a wake/break event, the system will wake up and an SCI will be generated.

25.12 USB Legacy Keyboard Operation

When a USB keyboard is plugged into the system, and a standard keyboard is not, the system may not boot, and DOS legacy software will not run, because the keyboard will not be identified. The Intel® 3100 Chipset implements a series of trapping operations which will snoop accesses that go to the keyboard controller, and put the expected data from the USB keyboard into the keyboard controller.

The following table summarizes the implementation of the bits in the USB Legacy Keyboard/Mouse Control Registers.

Table 746. USB Legacy Keyboard/Mouse Control Register Bit Implementation (Sheet 1 of 2)

Bit #	Bit Name	Summary	Details
15	SMI Caused by End of Pass-Through	Logically 1 bit for all controllers	This bit in all host controllers will be set at the same time and cleared at the same time. It is cleared whenever software writes a 1 to this bit in any of the classic USB host controllers. This bit may either be implemented separately for each controller or shared and aliased.
13	PCI Interrupt Enable	Independent enable	Each bit provides individual host control
12	SMI Caused by USB Interrupt	Independent status	Individual status bits for each controller
11	SMI Caused by Port 64 Write	Logically 1 bit for all controllers	This bit in all host controllers will be set at the same time and cleared at the same time. It is cleared whenever software writes a 1 to this bit in any of the classic USB host controllers. This bit may either be implemented separately for each controller or shared and aliased.

Table 746. USB Legacy Keyboard / Mouse Control Register Bit Implementation (Sheet 2 of 2)

Bit #	Bit Name	Summary	Details
10	SMI Caused by Port 64 Read	Logically 1 bit for all controllers	This bit in all host controllers will be set at the same time and cleared at the same time. It is cleared whenever software writes a 1 to this bit in any of the classic USB host controllers. This bit may either be implemented separately for each controller or shared and aliased.
9	SMI Caused by Port 60 Write	Logically 1 bit for all controllers	This bit in all host controllers will be set at the same time and cleared at the same time. It is cleared whenever software writes a 1 to this bit in any of the classic USB host controllers. This bit may either be implemented separately for each controller or shared and aliased.
8	SMI Caused by Port 60 Read	Logically 1 bit for all controllers	This bit in all host controllers will be set at the same time and cleared at the same time. It is cleared whenever software writes a 1 to this bit in any of the classic USB host controllers. This bit may either be implemented separately for each controller or shared and aliased.
7	SMI at End of Pass-Through Enable	Separate enables ORed together	This bit enables the generation of the SMI based on bit 15 within the same function. If bit 15 is implemented as a shared/aliased bit across all functions, then the bit 7s from all classic USB controllers are ORed together and used to enable the SMI based on bit 15.
6	Pass Through State	Logically 1 bit for all controllers	This bit in all host controllers reflects the state of the Pass-Through state machine. Software can force this bit to 0 by clearing the A20Gate Pass-Through Enable (bit 5) in *all* of the host controllers.
5	A20Gate Pass-Through Enable	ORed together to enable the pass-thru state machine	If any of these bits in the classic USB host controllers is set, then the Intel® 3100 Chipset will enable the Legacy Keyboard A20Gate Pass-through sequence. This prevents the SMI status bits (11:8) from asserting in all controllers when the specific sequence of I/O cycles is observed.
4	SMI on USB IRQ	Independent Enable	Each bit provides individual host control
3	SMI on Port 64 Writes Enable	Separate enables ORed together	Each bit enables SMI generation if the corresponding bit 11 is set. If bit 11 is implemented as a shared/aliased bit across all functions, then the bit 3s from all classic USB controllers are ORed together and used to enable the SMI based on bit 11.
2	SMI on Port 64 Reads Enable	Separate enables ORed together	Each bit enables SMI generation if the corresponding bit 10 is set. If bit 10 is implemented as a shared/aliased bit across all functions, then the bit 2s from all classic USB controllers are ORed together and used to enable the SMI based on bit 10.
1	SMI on Port 60 Writes Enable	Separate enables ORed together	Each bit enables SMI generation if the corresponding bit 9 is set. If bit 9 is implemented as a shared/aliased bit across all functions, then the bit 1s from all classic USB controllers are ORed together and used to enable the SMI based on bit 9.
0	SMI on Port 60 Reads Enable	Separate enables ORed together	Each bit enables SMI generation if the corresponding bit 8 is set. If bit 8 is implemented as a shared/aliased bit across all functions, then the bit 0s from all classic USB controllers are ORed together and used to enable the SMI based on bit 8.

Note: The scheme described below assumes that the keyboard controller (8042 or equivalent) is on the LPC bus.

This legacy operation is performed through SMM space. The latched SMI source (60R, 60W, 64R, 64W) is available in the Status Register. Because the enable is after the latch, it is possible to check for other events that didn't necessarily cause an SMI. It is the software's responsibility to logically AND the value with the appropriate enable bits.

Note: SMI is generated before the LPC cycle completes on the Intel® 3100 Chipset interface to ensure that the processor doesn't complete the cycle before the SMI is observed. The logic will also need to block the accesses to the 8042.

If there is an external 8042, then this is simply accomplished by not activating the 8042 CS. This is simply done by logically ANDing the four enables (60R, 60W, 64R, 64W) with the 4 types of accesses to determine if 8042CS should go active. An additional term is required for the “pass-through” case.

26.0 Device 31, Function 3: SMBus Controller Functional Description

26.1 Overview

The Intel® 3100 Chipset's IICH contains an SMBus Host interface that allows the processor to communicate with SMBus slaves.

Table 747 lists the SMBus signals and the actions taken during various power events

Table 747. SMBus signals

Signal Name	Power Plane	During Reset	After Reset	S3	S5	Alt Driver
SMBDATA	Resume	See note		High-Z	High-Z	Peripherals
SMBCLK	Resume			High-Z	High-Z	Peripherals
SMBALERT#	Resume	Can be driven high or low.				Peripherals

Note: SMBDATA and SMBCLK might go active if other devices are using the bus.

Unless specified, all of SMBus logic and registers in this chapter are reset by either CF9 reset or RSMRST#.

26.1.1 Host Controller

The Intel® 3100 Chipset provides an SMBus 2.0-compliant host controller. The host controller provides a mechanism for the processor to initiate communications with SMB peripherals (slaves). The Intel® 3100 Chipset is also capable of operating in a mode in which it can communicate with I^2C compatible devices.

The Intel® 3100 Chipset can perform SMBus messages with either PEC enabled or disabled. The actual PEC calculation and checking is performed in software. The SMBus Host Controller logic can automatically append the CRC byte if configured to do so.

The Intel® 3100 Chipset SMBus logic exists in Device 31, Function 3 configuration space, and consists of a transmit data path and host controller. The transmit data path provides the data flow logic needed to implement the seven different SMB command protocols and is controlled by the host controller. The logic is clocked by the RTC clock.

The SMBus Address Resolution Protocol (ARP) is supported through software by using the existing host controller commands, except for the new Host Notify command (which is actually a received message).

The programming model of the host controller is combined into two portions: A PCI configuration portion and a system I/O mapped portion. All static configuration, such as the I/O base address, is done via the PCI configuration space. Real-time programming of the host interface is done in system I/O space.

The Host Controller needs to check for parity errors as a target. If it sees an error, it must set the detected parity error bit (bit 15 of status). If bit 6 and bit 8 of the command register are set, it needs to generate SERR#, and set the signalled SERR# bit in the status register (bit 14).

26.1.2 Slave Interface

The slave interface allows an external master to write or read. The write cycles can be used to cause certain events or pass messages, and the read cycles can be used to determine the state of various status bits. The internal Host Controller cannot access the internal Slave Interface.

26.2 SMBus Controller Configuration Registers

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 748. Function 3 Configuration Registers Summary

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00	01h	VID	Vendor ID Register	8086h	RO
02	03h	DID	Device ID Register	269Bh	RO
04	05h	CMD	Command Register	0000h	RW
06	07h	DS	Device Status Register	0280h	RW
08	08h	RID	Revision ID Register	See Desc.	RO
09	09h	PI	Programming Interface Register	00h	RO
0A	0Ah	SCC	Sub Class Code Register	05h	RO
0B	0Bh	BCC	Base Class Code Register	0Ch	RO
20	23h	SBA	Base Address Register	00000001h	RW
2C	2Dh	SVID	SVID Register	0000h	RO
2E	2Fh	SID	SID Register	0000h	RO
3C	3Ch	INTLN	Interrupt Line Register	00h	RW
3D	3Dh	INTPN	Interrupt Pin Register	See Desc.	RO
40	40h	HCFG	Host Configuration Register	00h	RW, RO
F8	FBh	MANID	Manufacturer's ID Register	00010F80h	RO

Notes:
1. Registers not listed in Table 748 are reserved. These will return 00h on reads and writes have no effect.
2. All the above registers are implemented in the core well.
3. The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

26.2.1 Register Descriptions

26.2.1.1 Offset 00 - 01h: VID – Vendor ID Register

Table 749. Offset 00 - 01h: VID – Vendor ID Register

Device: 31	***Function:*** 3
Offset: 00 - 01h	***Size:*** 16 bit
Default Value: 8086h	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
15:0	VID	**Vendor ID:** This is a 16-bit value assigned to Intel	8086h	RO

26.2.1.2 Offset 02 - 03h: DID – Device ID Register

Table 750. Offset 02 - 03h: DID – Device ID Register

Device: 31	***Function:*** 3
Offset: 02 - 03h	***Size:*** 16 bit
Default Value: 269Bh	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
15:00	DID	**Device ID:** Indicates the device number assigned by the SIG.	269Bh	RO

26.2.1.3 Offset 04 - 05h: CMD – Command Register

Table 751. Offset 04 - 05h: CMD – Command Register (Sheet 1 of 2)

Device: 31	***Function:*** 3
Offset: 04 - 05h	***Size:*** 16 bit
Default Value: 0000h	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
15:11	Reserved	Reserved	0h	
10	INTD	**Interrupt Disable:** 0 = Enable (default) 1 = Disables SMBus to assert its PIRQB# signal	0b	RW
09	FBE	**Fast Back to Back Enable:** Reserved as '0'.	0b	
08	SERR_EN	**SERR# Enable:** 0 = Disables SERR# generation 1 = Enables SERR# generation	0b	RW
07	WCC	**Wait Cycle Control:** Reserved as '0'.	0b	

Table 751. Offset 04 - 05h: CMD – Command Register (Sheet 2 of 2)

Device: 31	**Function:** 3
Offset: 04 - 05h	**Size:** 16 bit
Default Value: 0000h	**Power Well:** Resume

Bits	Name	Description	Reset Value	Access
06	PER	**Parity Error Response:** 0 = Disable 1 = Sets Detected Parity Error bit (D3, F3, 06, bit 15) when a parity error is detected.	0b	RW
05:01	Reserved	Reserved	00h	
00	IOSE	**I/O Space Enable:** 0 = Disables access to the SM Bus I/O space registers as defined by the Base Address Register 1 = Enables access to the SM Bus I/O space registers as defined by the Base Address Register	0b	RW

26.2.1.4 Offset 06 - 07h: DS – Device Status Register

Table 752. Offset 06 - 07h: DS – Device Status Register

Device: 31	**Function:** 3
Offset: 06 - 07h	**Size:** 16 bit
Default Value: 0280h	**Power Well:** Resume

Bits	Name	Description	Reset Value	Access
15	DPE	**Detected Parity Error:** 0 = No parity error detected 1 = Parity error detected	0	RWC
14	SSE	**Signaled System Error:** 0 = No system error detected 1 = System error detected	0	RWC
13	RMA	**Received Master Abort:** Reserved as ‘0’.	0	
12	RTA	**Received Target Abort:** Reserved as ‘0’.	0	
11	STA	**Signaled Target-Abort Status:** 0 = Did not terminate transaction for this function with a target abort 1 = The function is targeted with a transaction that terminates with a target abort	0	RW
10:09	DEVT	**DEVSEL# Timing Status:** This 2-bit field defines the timing for DEVSEL# assertion. These read-only bits indicate the DEVSEL# timing when performing a positive decode. ***Note:*** The Intel® 3100 Chipset generates DEVSEL# with medium time. ***Note:*** It is not clear if a PCI master can write to SMBus controller.	01	RO
08	Reserved	Reserved	0	
07	Reserved	Reserved	1	
06:05	Reserved	Reserved	0	
04	CAP_LIST	**Capabilities List Indicator:** Hardwired to ‘0’ because there are no capability list structures in this function	0	RO
03	INTS	**Interrupt Status:** This bit indicates that an interrupt is pending. It is independent from the state of the Interrupt Enable bit in the command register.	0	RO
02:00	Reserved	Reserved.	00	

26.2.1.5 Offset 08h: RID – Revision ID Register

The value reported in this register depends on the value written to the Revision ID in Device 31, Function 0, Offset 08h. See Chapter 16.0, "Device 31, Function 0: LPC Interface," for details.

Table 753. Offset 08h: RID – Revision ID Register

Device: 31	***Function:*** 3
Offset: 08h	***Size:*** 8 bit
Default Value: See Desc	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
7:0	RID	**Revision Identifier:** Indicates the revision identifier. The value reported in this register depends on the value written to the Revision ID in Device 31, Function 0, Offset 08h.	See Desc	RO

26.2.1.6 Offset 09h: PI – Programming Interface Register

Table 754. Offset 09h: PI – Programming Interface Register

Device: 31	***Function:*** 3
Offset: 09h	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
7:0	PI	Reserved	00h	

26.2.1.7 Offset 0Ah: SCC – Sub Class Code Register

A value of 05h indicates that this device is a SM Bus serial controller.

Table 755. Offset 0Ah: SCC – Sub Class Code Register

Device: 31	***Function:*** 3
Offset: 0Ah	***Size:*** 8 bit
Default Value: 05h	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
7:0	SCC	**Sub Class Code:** 05h = IICH SM Bus serial controller	05h	RO

26.2.1.8 Offset 0Bh: BCC – Base Class Code Register

A value of 0Ch indicates that this device is a serial controller.

Table 756. Offset 0Bh: BCC – Base Class Code Register

Device: 31	***Function:*** 3
Offset: 0Bh	***Size:*** 8 bit
Default Value: 0Ch	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
7:0	BCC	**Base Class Code:** 0Ch = Serial controller.	0Ch	RO

26.2.1.9 Offset 20h - 23h: SMB_BASE – SMB Base Address Register

Table 757. Offset 20h - 23h: SMB_BASE – SMB Base Address Register

Device: 31	***Function:*** 3
Offset: 20 - 23h	***Size:*** 32 bits
Default Value: 00000001h	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0	
15:05	BASE_AD	**Base Address:** Provides the 32 byte system I/O base address for the SMB logic.	0	RW
04:01	Reserved	Reserved	0	
00	IOSI	**I/O Space Indicator:** This read-only bit always is 1, indicating that the SMB logic is I/O mapped.	1	RO

26.2.1.10 Offset 2C - 2Dh: SVID – SVID Register

BIOS sets the value in this register to identify the Subsystem Vendor ID. The SMBus SVID register, in combination with the SMBus Subsystem ID register, enables the operating system to distinguish each subsystem from the others.

Note: The software can write to this register only once per core well reset. Writes must be done as a single 16-bit cycle.

Table 758. Offset 2C - 2Dh: SVID – SVID Register

Device: 31	***Function:*** 3
Offset: 2C - 2Dh	***Size:*** 16 bit
Default Value: 0000h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:0	SVID	**Subsystem Vendor ID:** The SVID register, in combination with the Subsystem ID (SID) register, enables the operating system (OS) to distinguish subsystems from each other. ***Note:*** Software can write to this register only once per core well reset. Writes must be done as a single 16-bit cycle.	00h	RWO

26.2.1.11 Offset 2E - 2Fh: SID – Subsystem Identification Register

BIOS sets the value in this register to identify the Subsystem ID. The SMBus SID register, in combination with the SMBus SVID register, enables the operating system to distinguish each subsystem from the others.

Note: The software can write to this register only once per core well reset. Writes must be done as a single 16-bit cycle.

Table 759. Offset 2E - 2Fh: SID – Subsystem Identification Register

Device: 31	***Function:*** 3
Offset: 2E - 2Fh	***Size:*** 16 bit
Default Value: 0000h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:0	SID	**Subsystem ID:** The SID register, in combination with the SVID register, enables the operating system (OS) to distinguish subsystems from each other. ***Note:*** Software can write to this register only once per core well reset. Writes must be done as a single 16-bit cycle	00h	RWO

26.2.1.12 Offset 3Ch: INTLN – Interrupt Line Register

Table 760. Offset 3Ch: INTLN – Interrupt Line Register

Device: 31	***Function:*** 3
Offset: 3Ch	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
07:00	INTLN	**Interrupt line:** This data is not used. It is used to communicate to software the interrupt line that the interrupt pin is connected to PIRQB#.	00h	RW

26.2.1.13 Offset 3Dh: INTPN – Interrupt Pin Register

Table 761. Offset 3Dh: INTPN – Interrupt Pin Register

Device: 31	***Function:*** 3
Offset: 3Dh	***Size:*** 8 bit
Default Value: See Description	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
07:00	INTPN	**Interrupt Pin:** This reflects the value of D31IP.SMIP in the Intel® 3100 Chipset configuration space.	See Desc	RO

26.2.1.14 Offset 40h: HCFG – Host Configuration Register

Table 762. Offset 40h: HCFG – Host Configuration Register

Device: 31	***Function:*** 3
Offset: 40h	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
07:03	Reserved	Reserved	00h	
02	I2C_EN	0 = SMBus behavior 1 = Enabled to communicate with I^2C devices. This changes the formatting of some commands.	0	RW
01	SMB_SMI_EN	0 = SMBus interrupts will not generate an SMI# 1 = Any source of an SMB interrupt is instead be routed to generate an SMI#. Refer to Section 26.7 (Interrupts / SMI#). This bit needs to be set for SMBALERT# to be enabled.	0	RW
00	HST_EN	0 = Disable the SMBus Host controller 1 = Enable. The SMB Host controller interface is enabled to execute commands. The INTREN bit (offset SMBASE + 02h, bit 0) needs to be enabled for the SMB Host controller to interrupt or SMI#. The SMB Host controller does not respond to any new requests until all interrupt requests have been cleared.	0	RW

26.2.1.15 Offset F8 - FBh: MANID – Manufacturer's ID Register

Table 763. Offset F8 - FBh: MANID – Manufacturer's ID Register

Device: 31	***Function:*** 3
Offset: F8 - FBh	***Size:*** 32 bits
Default Value: 00010F80h	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
31:24	Reserved	Reserved	00h	
23:16	SID	**Stepping Identifier:** This field increments for each stepping of the part. This field can be used by software to differentiate steppings when the Revision ID may not change. See Section 16.2.8.1, "Offset F8h: MANID – Manufacturer's ID Register" on page 595 for cases in which the Revision ID may not increment.	01h - A1	RO
15:08	MID	**Manufacturing Identifier:** Indicates 0Fh = Intel	0Fh	RO
07:00	Reserved	Reserved	80h	

26.3 I/O Registers

Table 764. SMB I/O Registers Summary (Sheet 1 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	00h	HSTS	Host Status Register	00h	RWC
02h	02h	HCTL	Host Control Register	00h	RW
03h	03h	HCMD	Host Command Register	00h	RW
04h	04h	TSA	Transmit Slave Address Register	00h	RW
05h	05h	HD0	Host Data 0 Register	00h	RW

Table 764. SMB I/O Registers Summary (Sheet 2 of 2)

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
06h	06h	HD1	Host Data 1 Register	00h	RW
07h	07h	HBD	Host Block Data Register	00h	RW
08h	08h	PEC	Packet Error Chec Register	00h	RW
09h	09h	RSA	Receive Slave Address Register	44h	RW
0Ah	0Bh	SD	Slave Data Register	0000h	RO
0Ch	0Ch	AUXS	Auxiliary Status Register	00h	RW
0Dh	0Dh	AUXC	Auxiliary Control Register	00h	RW
0Eh	0Eh	SLPC	SM Link Pin Control <TCO Compatible mode only> Register	See Description	RW
0Fh	0Fh	SMBC	SM Bus Pin Control Register	See Description	RW
10h	10h	SSTS	Slave Status Register	00h	RWC
11h	11h	SCMD	Slave Command Register	00h	RW
14h	14h	NDA	Notify Device Address Register	00h	RO
16h	16h	NDLB	Notify Data Low Byte Register	00h	RO
17h	17h	NDHB	Notify Data High Byte Register	00h	RO

26.3.1 Register Descriptions

26.3.1.1 Offset 00h: HSTS – Host Status Register

All status bits are set by hardware and cleared by the software writing a one to the particular bit position. Writing a zero to any bit position has no effect.

Table 765. Offset 00h: HSTS – Host Status Register (Sheet 1 of 2)

I/O Address: SMBASE + 00h ***Size:*** *8 bit*

Default Value: 00h ***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
07	BDS	**BYTE_DONE_STS:** 0 = Software can clear this by writing a 1 1 = Host controller received a byte (for Block Read commands) or if it has completed transmission of a byte (for Block Write commands) when the 32-byte buffer is not being used. This bit is set, even on the last byte of the transfer. This bit is not set when transmission is due to the LAN interface heartbeat. This bit has no meaning for block transfers when the 32-byte buffer is enabled. When the last byte of a block message is received, the host controller sets this bit. However, it does not immediately set the INTR bit (bit 1 in this register). When the interrupt handler clears the BYTE_DONE_STS bit, the message is considered complete, and the host controller will then set the INTR bit (and generate another interrupt). Thus, for a block message of n bytes, n+1 interrupts will be generated. The interrupt handler needs to be implemented to handle these cases.	0	RWC
06	IUS	**In Use Status:** 0 = After a full PCI reset, a read to this bit returns a 0. 1 = After the first read, subsequent reads return a 1. A write of a 1 to this bit resets the next read value to 0. Writing a 0 to this bit has no effect. Software can poll this bit until it reads a 0, and then own the usage of the host controller. This bit has no other effect on the hardware, and is only used as semaphore among various independent software threads that may need to use the SMBus logic.	0	RW

Table 765. Offset 00h: HSTS – Host Status Register (Sheet 2 of 2)

I/O Address: SMBASE + 00h	***Size:*** *8 bit*
Default Value: 00h	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
05	SMBALERT_STS	**System Bus Alert Status:** 0 = Interrupt or SMI# was not generated by SMBALERT#. Software clears this bit by writing a 1 to it. 1 = The source of the interrupt or SMI# was the SMBALERT# signal. This bit is only cleared by software writing a 1 to the bit position or by CF9 RESET or RSMRST# going low (but not PLTRST#). If the signal is programmed as a GPI, then this bit is never set.	0	RWC
04	FAIL	**Failed:** 0 = Software clears this bit by writing a 1 to it. 1 = The source of the interrupt or SMI# was a failed bus transaction. This bit is set in response to the KILL bit being set to terminate the host transaction.	0	RWC
03	MCERR	**Machine Check Error:** 0 = Software clears this bit by writing a 1 to it. 1 = The source of the interrupt of SMI# was a transaction collision.	0	RWC
02	DERR	**Device Error:** 0 = Software clears this bit by writing a 1 to it, then deasserts the interrupt or SMI#. 1 = The source of the interrupt or SMI# was due to one of the following: • Illegal Command Field· • Unclaimed Cycle (host initiated) • Host Device Time-out Error • CRC Error	0	RWC
01	INTR	**Interrupt:** When set, this indicates that the source of the interrupt or SMI# was the successful completion of its last command.	0	RWC
00	HBSY	**Host Busy:** 0 = Cleared when the current transaction is completed 1 = Indicates that the Intel® 3100 Chipset is running a command from the host interface. No SMB registers must be accessed while this bit is set, except the BLOCK DATA BYTE Register. The BLOCK DATA BYTE Register can be accessed when this bit is set only when the SMB_CMD bits in the Host Control Register are programmed for Block command or I^2C Read command. This is necessary in order to check the DONE_STS bit.	0	RWC

26.3.1.2 Offset 02h: HCTL – Host Control Register

Table 766. Offset 02h: HCTL – Host Control Register (Sheet 1 of 3)

I/O Address: SMBASE + 02h ***Size:*** 8 bit

Default Value: 00h ***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
07	PEC_EN	**Packet Error Check Enable:** 0 = SMBus host controller does not perform the transaction with the PEC phase appended. 1 = Causes the host controller to perform the SMBus transaction with the Packet Error Checking phase appended. For writes, the value of the PEC byte is transferred from the PEC Register. For reads, the PEC byte is loaded in to the PEC Register. This bit must be written prior to the write in which the start bit is set.	0	RW
06	START	0 = This bit will always return 0 on reads. The HOST_BUSY bit in the Host Status register (offset 00h) can be used to identify when the command is finished. 1 = Writing a 1 to this bit initiates the command described in the SMB_CMD field. All registers should be setup prior to writing a 1 to this bit position.	0	RW
05	LAST_BYTE	Used for I^2C Read commands as an indication that the next byte is the last one to be received for that block. The algorithm and usage model for this bit is as follows (assume a message of n bytes): 1. When the software sees the BYTE_DONE_STS bit set (bit 7 in the SMBus Host Status Register) for each of bytes 1 through n-2 of the message, the software should then read the Block Data Byte Register to get the byte that was just received. 2. After reading each of bytes 1 to n-2 of the message, the software will then clear the BYTE_DONE_STS bit. 3. After receiving byte n-1 of the message, the software will then set the "LAST BYTE" bit. The software will then clear the BYTE_DONE_STS bit. 4. The Intel® 3100 Chipset then receives the last byte of the message (byte n). However, the state machine sees the LAST BYTE bit set, and instead of sending an ACK after receiving the last byte, it instead sends a NAK. 5. After receiving the last byte (byte n), the software still clears the BYTE_DONE_STS bit. However, the LAST_BYTE bit is irrelevant at that point. ***Notes:*** 1. This bit may be set when the TCO timer causes the SECOND_TO_STS bit to be set. See section Section 23.2.2.6, bit 1 for more details on that bit. The SMBus device driver should clear the LAST_BYTE bit (if it is set) before starting any new command. 2. In addition to I^2C Read Commands, the LAST_BYTE bit also causes Block Read/Write cycles to stop prematurely (at the end of the next byte).	0	RW

Note: A read to this register clears the pointer in the 32-byte buffer.

Table 766. Offset 02h: HCTL – Host Control Register (Sheet 2 of 3)

I/O Address: SMBASE + 02h ***Size:*** 8 bit

Default Value: 00h ***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
04:02	SMB_CMD	As shown by the bit encoding below, indicates which command is to be performed. If enabled, the Intel® 3100 Chipset generates an interrupt or SMI# when the command has completed If the value is for a non-supported or reserved command, the Intel® 3100 Chipset will sets the device error (DEV_ERR) status bit and generates an interrupt when the start bit is set. The Intel® 3100 Chipset performs no command, and does not operate until DEV_ERR is cleared. Bits / Name / Command Description 000 / Quick / The slave address and read/write value (bit 0) are stored in the tx slave address register. 001 / Byte / This command uses the transmit slave address and command registers. Bit 0 of the slave address register determines if this is a read or write command. 010 / Byte Data / This command uses the transmit slave address, command, and DATA0 registers. Bit 0 of the slave address register determines if this is a read or write command. If it is a read, the DATA0 register contains the read data. 011 / Word Data / This command uses the transmit slave address, command, DATA0 and DATA1 registers. Bit 0 of the slave address register determines if this is a read or write command. If it is a read, after the command completes the DATA0 and DATA1 registers contain the read data. 100 / Process Call / This command uses the transmit slave address, command, DATA0 and DATA1 registers. Bit 0 of the slave address register determines if this is a read or write command. After the command completes, the DATA0 and DATA1 registers will contain the read data. 101 / Block / This command uses the transmit slave address, command, and DATA0 registers, and the Block Data Byte register. For block write, the count is stored in the DATA0 register and indicates how many bytes of data will be transferred. For block reads, the count is received and stored in the DATA0 register. Bit 0 of the slave address register selects if this is a read or write command. For writes, data is retrieved from the first n (where n is equal to the specified count) addresses of the SRAM array. For reads, the data is stored in the Block Data Byte register. 110 / I^2C Read / This command uses the transmit slave address, command, DATA0, DATA1 registers, and the Block Data Byte register. The read data is stored in the Block Data Byte register. Intel® 3100 Chipset will continue reading data until the NAK is received.	000	RW

Note: A read to this register clears the pointer in the 32-byte buffer.

Table 766. Offset 02h: HCTL – Host Control Register (Sheet 3 of 3)

I/O Address: SMBASE + 02h			***Size:*** 8 bit	
Default Value: 00h			***Power Well:*** Resume	
Bits	**Name**	**Description**	**Reset Value**	**Access**
04:02 (cont'd)	SMB_CMD (cont'd)	Bits / Name / Command Description 111 / Block-Process / This command uses the transmit slave address, command, DATA0 and the Block Data Byte register. For block write, the count is stored in the DATA0 register and indicates how many bytes of data will be transferred. For block read, the count is received and stored in the DATA0 register. Bit 0 of the slave address register always indicate a write command. For writes, data is retrieved from the first m (where m is equal to the specified count) addresses of the SRAM array. For reads, the data is stored in the Block Data Byte register. ***Note:*** E32B bit in the Auxiliary Control Register must be set for this command to work.	000	RW
01	KILL	0 = Normal SMBus host controller functionality. 1 = Kills the current host transaction taking place, sets the FAILED status bit, and asserts the interrupt (or SMI#). This bit, once set, must be cleared by software to allow the SMBus host controller to function normally.	0	RW
00	INTREN	0 = Disable 1 = Enable the generation of an interrupt or SMI# upon the completion of the command	0	RW

Note: A read to this register clears the pointer in the 32-byte buffer.

26.3.1.3 Offset 03h: HCMD – Host Command Register

This eight bit field is transmitted by the host controller in the command field of the SMB protocol during the execution of any command.

Table 767. Offset 03h: HCMD – Host Command Register

I/O Address: SMBASE + 03h			***Size:*** 8 bit	
Default Value: 00h			***Power Well:*** Resume	
Bits	**Name**	**Description**	**Reset Value**	**Access**
7:0	HST_CMD	This 8-bit field is transmitted by the host controller in the command field of the SMBus protocol during the execution of any command.	00h	RW

26.3.1.4 Offset 04h: TSA – Transmit Slave Address Register

This register is transmitted by the host controller in the slave address field of the SMB protocol. This is the address of the target.

Table 768. Offset 04h: TSA – Transmit Slave Address Register

I/O Address: SMBASE + 04h	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
07:01	ADDRESS	7-bit address of the targeted slave	0000000	RW
00	RW	Direction of the host transfer. 0 = write 1 = read	0	RW

26.3.1.5 Offset 05h: HD0 – Data 0 Register

Table 769. Offset 05h: HD0 – Data 0 Register

I/O Address: SMBASE + 05h	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
07:00	DATA0/ COUNT	This field contains the eight bit data sent in the DATA0 field of the SMB protocol. For block write commands, this register reflects the number of bytes to transfer. This register should be programmed to a value between 1 and 32 for block counts. A count of 0 or a count above 32 will result in unpredictable behavior. The host controller does not check or log illegal block counts.	00h	RW

26.3.1.6 Offset 06h: HD1 – Data 1 Register

Table 770. Offset 06h: HD1 – Data 1 Register

I/O Address: SMBASE + 06h	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Resume

Bits	Name	Description	Reset Value	Access
07:00	DATA1	This eight bit register is transmitted in the DATA1 field of the SMB protocol during the execution of any command.	00h	RW

26.3.1.7 Offset 07h: HBD – Host Block Data Register

Table 771. Offset 07h: HBD – Host Block Data Register

I/O Address: SMBASE + 07h			***Size:*** 8 bit	
Default Value: 00h			***Power Well:*** Resume	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:00	BDTA	**Block Data:** This is either a register, or a pointer into a 32-byte block array, depending upon whether the E32B bit is set in the Auxiliary Control register. When the E32B bit is cleared, this is a register containing a byte of data to be sent on a block write or read from on a block read, just as it behaved on the Intel® 3100 Chipset. When the E32B bit is set, reads and writes to this register are used to access the 32-byte block data storage array. An internal index pointer is used to address the array, which is reset to 0 by reading the HCTL register (offset 02h). The index pointer then increments automatically upon each access to this register. The transfer of block data into (read) or out of (write) this storage array during an SMBus transaction always starts at index address 0. When the E2B bit is set, for writes, software will write up to 32-bytes to this register as part of the setup for the command. After the Host Controller has sent the Address, Command, and Byte Count fields, it will send the bytes in the SRAM pointed to by this register. When the E2B bit is cleared for writes, software will place a single byte in this register. After the host controller has sent the address, command, and byte count fields, it will send the byte in this register. If there is more data to send, software will write the next series of bytes to the SRAM pointed to by this register and clear the DONE_STS bit. The controller will then send the next byte. During the time between the last byte being transmitted to the next byte being transmitted, the controller will insert wait-states on the interface. When the E2B bit is set for reads, after receiving the byte count into the Data0 register, the first series of data bytes go into the SRAM pointed to by this register. If the byte count has been exhausted or the 32-byte SRAM has been filled, the controller will generate an SMI# or interrupt (depending on configuration) and set the DONE_STS bit. Software will then read the data. During the time between when the last byte is read from the SRAM to when the DONE_STS bit is cleared, the controller will insert wait-states on the interface.	00	RW

26.3.1.8 Offset 08h: PEC – Packet Error Check Data Register

This register contains the 8-bit CRC value that is used as the Packet Error Check on SMBus. For writes, this register is written by software prior to running the command. For reads, this register is read by software after the read command is completed on SMBus.

Table 772. Offset 08h: PEC – Packet Error Check Data Register

I/O Address: SMBASE + 08h			***Size:*** 8 bit	
Default Value: 00h			***Power Well:*** Resume	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:00	PEC_DATA	This 8-bit register is written with the SMBus PEC data prior to a write transaction. For read transactions, the PEC data is loaded from the SMBus into this register and is then read by software. Software must ensure that the INUSE_STS bit is properly maintained to avoid having this field overwritten by a write transaction following a read transaction.	00h	RW

Note: This register may reside in either the core well or the suspend well. To simplify the implementation, this register is in the suspend well with the suspend well version of PCI reset (URST33B).

26.3.1.9 Offset 0Ch: AUXS – Auxiliary Status Register

Table 773. Offset 0Ch: AUXS – Auxiliary Status Register

I/O Address: SMBASE + 0Ch			***Size:*** 8 bit	
Default Value: 00h			***Power Well:*** Core	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:02	Reserved	Reserved	0	
01	STCO	**SMBus TCO mode:** This is the status bit that reflects the strap setting of legacy TCO mode vs. Advanced TCO mode. 0 = Indicates that this bit is zero, since Advanced TCO mode is not supported. 1 = Indicates it is in the Advanced TCO mode. When cleared, it is in the legacy/compatible TCO mode.	0	RO
00	CRCE	**CRC Error:** 0 = Software clears this bit by writing a 1 to it. 1 = This bit is set if a received message contained a CRC error. When this bit is set, the DERR bit of the host status register is also set. This bit is set by the controller if a software abort occurs in the middle of the CRC portion of the cycle or an abort happens after the Intel® 3100 Chipset has received the final data bit transmitted by an external slave.	0	RWC

26.3.1.10 Offset 0Dh: AUXC – Auxiliary Control Register

Table 774. Offset 0Dh: AUXC – Auxiliary Control Register

I/O Address: SMBASE + 0Dh			***Size:*** 8 bit	
Default Value: 00h			***Power Well:*** Resume	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:02	Reserved	Reserved	0	
01	E32B	**Enable 32-byte Buffer:** 0 = The Host Block Data register is a pointer into a single register. 1 = When set, the Host Block Data register is a pointer into a 32-byte buffer. This enables the block commands to transfer or receive up to 32-bytes before the Intel® 3100 Chipset generates an interrupt.	0	RW
00	AAC	**Automatically Append CRC:** 0 = Does not automatically append the CRC 1 = Automatically appends the CRC This bit must not be changed during SM Bus transactions, or undetermined behavior results. It should be programmed only once during the lifetime of the function.	0	RW

26.3.1.11 Offset 0Eh: SMLC – SMLINK_PIN_CTL Register

This register is only applicable in the TCO compatible mode.

This register is in the resume well and is reset by CF9 RESET or RSMRST#.

Table 775. Offset 0Eh: SMLC – SMLINK_PIN_CTL Register

I/O Address: SMBASE + 0Eh			***Size:*** 8 bit	
Default Value: See below			***Power Well:*** Resume; reset by CF9 RESET or RSMRST#	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:03	Reserved	Reserved		
02	SMLINK_ CLK_CTL	This read/write bit has a default of 1. 0 = Drives the SMLINK[0] pin low, independent of what the other SMLINK logic would otherwise indicate for the SMLINK[0] pin. 1 = The SMLINK[0] pin is *Not* overdriven low. The other SMLINK logic controls the state of the pin.	1	RW
01	SMLINK[1]_ CUR_STS	This read-only bit has a default value that is dependent on an external signal level. This pin returns the value on the SMLINK[1] pin. This allows software to read the current state of the pin. 0 = Low 1 = High	1	RO
00	SMLINK[0]_ CUR_STS	This read-only bit has a default value that is dependent on an external signal level. This pin returns the value on the SMLINK[0] pin. 0 = Low 1 = High This allows software to read the current state of the pin.	0	RO

26.3.1.12 Offset 0Fh: SMBC – SMBUS_PIN_CTL Register

This register is in the resume well and is reset by CF9 RESET or RSMRST#.

Table 776. Offset 0Fh: SMBC – SMBUS_PIN_CTL Register

I/O Address: SMBASE + 0Fh	***Size:*** 8 bit
Default Value: See below	***Power Well:*** Resume; reset by CF9 RESET or RSMRST#

Bits	Name	Description	Reset Value	Access
07:03	Reserved	Reserved.	01	
02	SMBCLK_CTL	This bit has a default of 1. 0 = Drives the SMBCLK pin low, independent of what the other SMB logic would otherwise indicate for the SMBCLK pin. 1 = The SMBCLK pin is *Not* overdriven low. The other SMBus logic controls the state of the pin.	1	RW
01	SMBDATA_ CUR_STS	This bit has a default value that is dependent on an external signal level. This pin returns the value on the SMBDATA pin. This allows software to read the current state of the pin. 0 = Low 1 = High	1	RO
00	SMBCLK_ CUR_STS	This bit has a default value that is dependent on an external signal level. This pin returns the value on the SMBCLK pin. This allows software to read the current state of the pin. 0 = Low 1 = High	1	RO

26.4 Host Controller

26.4.1 Overview

The SMB Host Controller is used to send commands to other SMB slave devices. Software sets up the host controller with an address, command, and for writes, data and optionally PEC; and then tells the controller to start. When the controller has finished transmitting data on writes, or receiving data on reads, it generates an SMI# or interrupt, if enabled.

The host controller supports eight command protocols of the SMB interface (see the *SMBus Specification, Rev. 2.0*): Quick Command, Send Byte, Receive Byte, Write Byte/ Word, Read Byte/Word, Process Call, Block Read, Block Write and Block write-block read process call.

The SMB Host Controller requires that the various data and command fields be setup for the type of command to be sent. When software sets the START bit, the SMB Host Controller performs the requested transaction and interrupt the processor (or generate an SMI#) when its finished. Once a START command has been issued, the values of the “active registers” (Host Control, Host Command, Transmit Slave Address, Data0, Data1) should not be changed or read until the interrupt status bit (INTR) has been set (indicating the completion of the command). Any register values needed for computation purposes should be saved prior to issuing of a new command, as the SMB Host Controller will update all registers while completing the new command.

The Intel® 3100 Chipset supports slave functionality, including the Host Notify protocol, on the SMLink pins when in TCO compatible mode. Therefore, in order to be fully compliant with the *SMBus Specification* (which requires the Host Notify cycle), the SMLink and SMBus signals must be tied together externally.

Using the SMB Host Controller to send commands to the SMB slave port is not supported.

26.4.2 Command Protocols

In all of the following commands, the Host Status Register (offset 00h) is used to determine the progress of the command. While the command is in operation, the HOST_BUSY bit is set. If the command completes successfully, the INTR bit is set in the Host Status Register. If the device does not respond with an acknowledge, and the transaction times out, the DEV_ERR bit is set. If software sets the KILL bit in the Host Control Register while the command is running, the transaction stops and the FAILED bit is set after the Intel® 3100 Chipset forces a timeout. In addition, if the KILL bit is set during the CRC cycle, both the CRCE and DEV_ERR bits are also set. When the KILL bit is set, the Intel® 3100 Chipset aborts current transaction by asserting SMBCLK low for greater than the timeout period, asserts a STOP condition and then releases SMBCLK and SMBDATA. However, setting the KILL bit does not affect SMLINK or TCO transactions or causes the Intel® 3100 Chipset to force a timeout if it is not performing a transaction.

26.4.2.1 Quick Command

When programmed for a quick command, the Transmit Slave Address Register is sent. Table 777 shows the order. The PEC byte is never appended to the Quick Protocol. Software must force the PEC_EN bit to ‘0’ when performing the Quick Command for possible future enhancements. Also, Quick Command with I2C_EN set produces undefined results. Software must force the I2C_EN bit to 0 when running this command.

Table 777. Quick Protocol

Bit	Description
1	Start Condition
2–8	Slave Address - 7 bits
9	Read / Write Direction
10	Acknowledge from slave
11	Stop

26.4.2.2 Send Byte/Receive Byte

For the send byte command, the Transmit Slave Address and Device Command Registers are sent.

The Receive Byte is similar to a Send Byte, the only difference is the direction of data transfer. When programmed for the receive byte command, the Transmit Slave Address Register is sent. The data received is stored in the DATA0 register. The order sent/received without PEC is shown in Table 778. Send Byte/Receive Byte command with I2C_EN set produces undefined results. Software must force the I2C_EN bit to 0 when running this command.

Table 778. Send/Receive Byte Protocol without PEC (Sheet 1 of 2)

Send Byte Protocol		Receive Byte Protocol	
Bit	Description	Bit	Description
1	Start	1	Start
2 – 8	Slave Address - 7 bits	2 – 8	Slave Address - 7 bits

Table 778. Send/Receive Byte Protocol without PEC (Sheet 2 of 2)

9	Write	9	Read
10	Acknowledge from slave	10	Acknowledge from slave
11 – 18	Command code - 8 bits	11 – 18	Data byte from slave
19	Acknowledge from slave	19	NOT Acknowledge
20	Stop	20	Stop

The order sent/received, with PEC, is shown in Table 779.

Table 779. PEC Send/Receive Order

Send Byte Protocol		Receive Byte Protocol	
Bit	**Description**	**Bit**	**Description**
1	Start	1	Start
2 – 8	Slave Address - 7 bits	2 – 8	Slave Address - 7 bits
9	Write	9	Read
10	Acknowledge from slave	10	Acknowledge from slave
11 – 18	Command code - 8 bits	11 – 18	Data byte from slave
19	Acknowledge from slave	19	Acknowledge
20 – 27	PEC	20 – 27	PEC from slave
28	Acknowledge from slave	28	Not Acknowledge
29	Stop	29	Stop

26.4.2.3 Write Byte/Word

The first byte of a write byte/word access is the command code. The next one or two bytes are the data to be written. When programmed for a write byte/word command, the Transmit Slave Address, device command, and Data0 Registers are sent. In addition, the Data1 Register is sent on a write word command. The order of bits without PEC is shown in Table 780. Issuing a write byte/word command with I2C_EN set produces undefined results. Software must force the I2C_EN bit to 0 when running this command.

The order of bits with PEC is shown in Table 781.

Table 780. Write Byte/Word Protocol Without PEC (Sheet 1 of 2)

Write Byte Protocol		Write Word Protocol	
Bit	**Description**	**Bit**	**Description**
1	Start	1	Start
2 – 8	Slave Address - 7 bits	2 – 8	Slave Address - 7 bits
9	Write	9	Write
10	Acknowledge from slave	10	Acknowledge from slave
11 – 18	Command code - 8 bits	11 – 18	Command code - 8 bits
19	Acknowledge from slave	19	Acknowledge from slave
20 – 27	Data Byte - 8 bits	20 – 27	Data Byte Low - 8 bits
28	Acknowledge from Slave	28	Acknowledge from Slave

Table 780. Write Byte / Word Protocol Without PEC (Sheet 2 of 2)

29	Stop	29 – 36	Data Byte High - 8 bits
		37	Acknowledge from slave
		38	Stop

Table 781. PEC Bit Order

Write Byte Protocol		Write Word Protocol	
Bit	**Description**	**Bit**	**Description**
1	Start	1	Start
2 – 8	Slave Address - 7 bits	2 – 8	Slave Address - 7 bits
9	Write	9	Write
10	Acknowledge from slave	10	Acknowledge from slave
11 – 18	Command code - 8 bits	11 – 18	Command code - 8 bits
19	Acknowledge from slave	19	Acknowledge from slave
20 – 27	Data Byte - 8 bits	20 – 27	Data Byte Low - 8 bits
28	Acknowledge from Slave	28	Acknowledge from Slave
29 – 36	PEC	29 – 36	Data Byte High - 8 bits
37	Acknowledge from Slave	37	Acknowledge from slave
38	Stop	38 – 45	PEC
		46	Acknowledge from slave
		47	Stop

26.4.2.4 Read Byte / Word

Reading data is slightly more complicated than writing data. First a command to the slave device must be written. Then it must follow that command with a repeated start condition to denote a read from that device's address. The slave then returns one or two bytes of data.

When programmed for the read byte/word command, the transmit slave address and device command registers are sent. Data is received into the DATA0 on the read byte, and the DAT0 and DATA1 registers on the read word. The order sent and received with PEC disabled is shown in Table 782.

Read byte/word command with I2C_EN set produces undefined results. Software must force the I2C_EN bit to 0 when running this command.

Table 782. Read Byte / Word Protocol without PEC (Sheet 1 of 2)

Read Byte Protocol		Read Word Protocol	
Bit	**Description**	**Bit**	**Description**
1	Start	1	Start
2 – 8	Slave Address - 7 bits	2 – 8	Slave Address - 7 bits
9	Write	9	Write
10	Acknowledge from slave	10	Acknowledge from slave
11 – 18	Command code - 8 bits	11 – 18	Command code - 8 bits
19	Acknowledge from slave	19	Acknowledge from slave

Table 782. Read Byte / Word Protocol without PEC (Sheet 2 of 2)

20	Repeated Start	20	Repeated Start
21 – 27	Slave Address - 7 bits	21 – 27	Slave Address - 7 bits
28	Read	28	Read
29	Acknowledge from slave	29	Acknowledge from slave
30 – 37	Data from slave - 8 bits	30 – 37	Data Byte Low from slave - 8 bits
38	NOT acknowledge	38	Acknowledge
39	Stop	39 – 46	Data Byte High from slave - 8 bits
		47	NOT acknowledge
		48	Stop

The order sent and received with PEC enabled is shown in Table 783.

Table 783. Read Byte / Word Protocol with PEC

Read Byte Protocol		Read Word Protocol	
Bit	**Description**	**Bit**	**Description**
1	Start	1	Start
2 – 8	Slave Address - 7 bits	2 – 8	Slave Address - 7 bits
9	Write	9	Write
10	Acknowledge from slave	10	Acknowledge from slave
11 – 18	Command code - 8 bits	11 – 18	Command code - 8 bits
19	Acknowledge from slave	19	Acknowledge from slave
20	Repeated Start	20	Repeated Start
21 – 27	Slave Address - 7 bits	21 – 27	Slave Address - 7 bits
28	Read	28	Read
29	Acknowledge from slave	29	Acknowledge from slave
30 – 37	Data from slave - 8 bits	30 – 37	Data Byte Low from slave - 8 bits
38	Acknowledge	38	Acknowledge
39 – 46	PEC from slave	39 – 46	Data Byte High from slave - 8 bits
47	NOT acknowledge	47	Acknowledge
48	Stop	48 – 55	PEC from slave
		56	NOT acknowledge
		57	Stop

26.4.2.5 Process Call

The process call is named because a command sends data and waits for the slave to return a value dependent on that data. The protocol is simply a write word followed by a read word, but without a second command or stop condition.

When programmed for the process call command, the Intel® 3100 Chipset transmits the transmit address, device command, and DATA0 and DATA1 registers. Data received from the device is stored in the DATA0 and DATA1 registers. The value written into bit 0 of the Transmit Slave Address Register (SMBus Offset 04h) needs to programmed to 0.

Note: If the I2C_EN bit is set, then the command field is not sent.

The order sent with PEC disabled is shown in Table 784. The process call command with I2C_EN set and either the PEC_EN or AAC bit set produces undefined results. Software must either force the I2C_EN bit or both PEC_EN and AAC bits to 0 when running this command.

Table 784. Process Call Protocol without PEC

Bit	Description
1	Start
2 – 8	Slave Address - 7 bits
9	Write
10	Acknowledge from Slave
11 – 18	Command code - 8 bits (Skip if I2C_EN is set)
19	Acknowledge from slave (Skip if I2C_EN is set)
20 – 27	Data byte Low - 8 bits
28	Acknowledge from Slave
29 – 36	Data Byte High - 8 bits
37	Acknowledge from slave
38	Repeated Start
39 – 45	Slave Address - 7 bits
46	Read
47	Acknowledge from slave
48 – 55	Data Byte Low from slave - 8 bits
56	Acknowledge
57 – 64	Data Byte High from slave - 8 bits
65	NOT acknowledge
66	Stop

The order sent with PEC enabled is shown in Table 785.

Table 785. Process Call Protocol with PEC (Sheet 1 of 2)

Bit	Description
1	Start
2 – 8	Slave Address - 7 bits
9	Write
10	Acknowledge from Slave
11 – 18	Command code - 8 bits
19	Acknowledge from slave
20 – 27	Data byte Low - 8 bits
28	Acknowledge from Slave
29 – 36	Data Byte High - 8 bits
37	Acknowledge from slave
38	Repeated Start
39 – 45	Slave Address - 7 bits
46	Read

Table 785. Process Call Protocol with PEC (Sheet 2 of 2)

Bit	Description
47	Acknowledge from slave
48 – 55	Data Byte Low from slave - 8 bits
56	Acknowledge
57 – 64	Data Byte High from slave - 8 bits
65	Acknowledge
66 – 73	PEC from slave
74	NOT acknowledge
75	Stop

26.4.2.6 Block Read/Write

The Intel® 3100 Chipset contains a 32-byte buffer for read and write data which can be enabled by setting bit '1' of the Auxiliary Control register at offset 0Dh in I/O space, as opposed to a single byte of buffering. This 32-byte buffer is filled with write data before transmission and filled with read data on reception. In the Intel® 3100 Chipset, the interrupt is generated only after a transmission or reception of 32 bytes, or when the entire byte count has been transmitted/received.

The block write command with I2C_EN set and either the PEC_EN or AAC bit set produces undefined results. Software must either force the I2C_EN bit or both PEC_EN and AAC bits to 0 when running this command.

26.4.2.6.1 SM Bus Mode

The block write begins with a slave address and a write condition. After the command code the Intel® 3100 Chipset issues a byte count describing how many more bytes will follow in the message. If a slave had 20 bytes to send, the first byte would be the number 20 (14h), followed by 20 bytes of data. The byte count may not be 0. A block read or write is allowed to transfer a maximum of 32 data bytes.

When programmed for a block write command, the transmit slave address, device command, and Data0 (count) registers are sent. Data is then sent from the Block Data Byte register; the total data sent being the value stored in the Data0 Register. On block read commands, the first byte received is stored in the Data0 register, and the remaining bytes are stored in the Block Data Byte register.

26.4.2.6.2 I^2C Mode

The format of the command changes slightly for block commands if the I2C_EN bit is set. The Intel® 3100 Chipset still sends the number of bytes (on writes) or receive the number of bytes (on reads) indicated in the DATA0 register. However, it does not send the contents of the DATA0 register as part of the message.

The format of the command changes slightly for a block write if the I2C_EN bit is set. The Intel® 3100 Chipset still sends the number of bytes indicated in the DATA0 register. However, it does not send the contents of the DATA0 register as part of the message.

The protocol for the block write and block read without PEC is shown in Table 786.

Table 786. Block Read/Write Protocol without PEC

Block Write Protocol		Block Read Protocol	
Bit	**Description**	**Bit**	**Description**
1	Start	1	Start
2 – 8	Slave Address - 7 bits	2 – 8	Slave Address - 7 bits
9	Write	9	Write
10	Acknowledge from slave	10	Acknowledge from slave
11 – 18	Command code - 8 bits	11 – 18	Command code - 8 bits
19	Acknowledge from slave	19	Acknowledge from slave
20 – 27	Byte Count - 8 bits (skip this step if I2C_EN bit set)	20	Repeated Start
28	Acknowledge from Slave (skip this step if I2C_EN bit set)	21 – 27	Slave Address - 7 bits
29 – 36	Data Byte 1 - 8 bits	28	Read
37	Acknowledge from Slave	29	Acknowledge from slave
38 – 45	Data Byte 2 - 8 bits	30 – 37	Byte Count from slave - 8 bits
46	Acknowledge from slave	38	Acknowledge
...	Data Bytes / Slave Acknowledges...	39 – 46	Data Byte 1 from slave - 8 bits
...	Data Byte N - 8 bits	47	Acknowledge
...	Acknowledge from Slave	48 – 55	Data Byte 2 from slave - 8 bits
...	Stop	56	Acknowledge
		...	Data Bytes from slave/Acknowledge
		...	Data Byte N from slave - 8 bits
		...	NOT Acknowledge
		...	Stop

The protocol for the block write and block read with PEC is shown in Table 787.

The block write command with I2C_EN set and the PEC_EN bit set produces undefined results. Software must force the PEC_EN bit to 0 when running this command.

Table 787. Block Read/Write Protocol with PEC

Block Write Protocol		Block Read Protocol	
Bit	**Description**	**Bit**	**Description**
1	Start	1	Start
2 – 8	Slave Address - 7 bits	2 – 8	Slave Address - 7 bits
9	Write	9	Write
10	Acknowledge from slave	10	Acknowledge from slave
11 – 18	Command code - 8 bits	11 – 18	Command code - 8 bits
19	Acknowledge from slave	19	Acknowledge from slave
20 – 27	Byte Count - 8 bits	20	Repeated Start
28	Acknowledge from Slave	21 – 27	Slave Address - 7 bits
29 – 36	Data Byte 1 - 8 bits	28	Read
37	Acknowledge from Slave	29	Acknowledge from slave
38 – 45	Data Byte 2 - 8 bits	30 – 37	Byte Count from slave - 8 bits
46	Acknowledge from slave	38	Acknowledge
...	Data Bytes / Slave Acknowledges...	39 – 46	Data Byte 1 from slave - 8 bits
...	Data Byte N - 8 bits	47	Acknowledge
...	Acknowledge from Slave	48 – 55	Data Byte 2 from slave - 8 bits
...	PEC – 8 bits	56	Acknowledge
...	Acknowledge from Slave	...	Data Bytes from slave/Acknowledge
...	Stop	...	Data Byte N from slave - 8 bits
		...	Acknowledge
		...	PEC from slave – 8 bits
		...	NOT Acknowledge
		...	Stop

26.4.2.7 I²C Read

This command allows the Intel® 3100 Chipset to perform block reads to certain I²C devices, such as serial E2PROMs. The SMBus Block Read supports the 7-bit addressing mode only. However this doesn't allow access to devices that need to use the I²C "Combined Format" that has data bytes after the address. Typically these data bytes correspond to an offset (address) within the serial memory chips.

The I²C Read command with either PEC_EN or AAC bit set produces undefined results. Software must force both PEC_EN and AAC bits to 0 when running this command.

To support these devices, the Intel® 3100 Chipset implements an I²C Read command with the following format:

Table 788. I²C Read Command Formats (Sheet 1 of 2)

Bit	Description
1	Start
2 – 8	Slave Address - 7 bits
9	Write
10	Acknowledge from slave
11 – 18	Send DATA1 register

Table 788. I²C Read Command Formats (Sheet 2 of 2)

19	Acknowledge from slave
20	Repeated Start
21 – 27	Slave Address - 7 bits
28	Read
29	Acknowledge from slave
30 – 37	Data Byte 1 from slave - 8 bits
38	Acknowledge
39 – 46	Data Byte 2 from slave - 8 bits
47	Acknowledge
...	Data Bytes from slave/Acknowledge
...	Data Byte N from slave - 8 bits
...	NOT Acknowledge
...	Stop

The Intel® 3100 Chipset continues reading data from the peripheral until the NAK is received.

Note: This new command is supported independent of the setting of the I2C_EN bit.

Note: The value written into bit 0 of the Transmit Slave Address register (SMBus Offset 04h) must be 0.

26.4.2.8 Block Write-Block Read Process Call

The block write-block read process call is a two-part message. The call begins with a slave address and a write condition. After the command code the host issues a write byte count (M) that describes how many more bytes will be written in the first part of the message. If a master has six bytes to send, the byte count field will have the value 6 (0000 0110b), followed by the six bytes of data. The write byte count (M) cannot be zero.

The second part of the message is a block of read data beginning with a repeated start condition followed by the slave address and a read bit. The next byte is the read byte count (N), which may differ from the write byte count (M). The read byte count (N) cannot be zero.

The combined data payload must not exceed 32 bytes. The byte length restrictions of this process call are summarized as follows:

- $M \geq 1$ byte
- $N \geq 1$ byte
- $M + N \leq 32$ bytes

The read byte count does not include the PEC byte. The PEC is computed on the total message beginning with the first slave address and using the normal PEC computational rules. It is highly recommended that a PEC byte be used with the block write-block read process call. Software must do a read to the command register (offset 2h) to reset the 32 byte buffer pointer prior to reading the block data register.

Note: There is no STOP condition before the repeated START condition, and that a NACK signifies the end of the read transfer.

Note: E32B in the Auxiliary Control register must be set when using this protocol.

Table 789. Block Write-Block Read Process Call Protocol with/without PEC

Bit	Description
1	Start
2 – 8	Slave Address - 7 bits
9	Write
10	Acknowledge from Slave
11 – 18	Command code - 8 bits
19	Acknowledge from slave
20 – 27	Data Byte Count (M) - 8 bits
28	Acknowledge from Slave
29 – 36	Data Byte (1) - 8 bits
37	Acknowledge from slave
38 – 45	Data Byte (2) - 8 bits
46	Acknowledge from slave
...	...
	Data Byte (M) - 8 bits
	Acknowledge from slave
	Repeated Start
	Slave Address - 7 bits
	Read
	Acknowledge from slave
	Data Byte Count (N) from slave – 8 bits
	Acknowledge from master
	Data Byte (1) from slave – 8 bits
	Acknowledge from master
	Data Byte (2) from slave – 8 bits
	Acknowledge from master
...	...
	Data Byte Count (N) from slave – 8 bits
	Acknowledge from master (Skip if no PEC)
	PEC from slave (Skip if no PEC)
	NOT acknowledge
	Stop

26.4.3 I²C Behavior

When the I2C_EN bit is set, the Intel® 3100 Chipset SMBus logic is set to communicate with I²C devices. This forces the following changes:

1. The Process Call command skips the command code (and its associated acknowledge)
2. The Block Write command skips sending the byte count (DATA0)

In addition, the Intel® 3100 Chipset supports the I²C Read command. This is independent of the I2C_EN bit. When operating in I²C mode, (I2C_EN bit set), the Intel® 3100 Chipset never uses the 32-byte buffer for any block commands.

26.4.4 Heartbeat for Use with External LAN

This method allows the Intel® 3100 Chipset to send messages to an External LAN controller when the processor is otherwise unable to do so. It uses the SMLink Interface between the Intel® 3100 Chipset and external LAN controller in TCO compatible mode. The actual heartbeat message is a block write. Only eight bytes are sent.

See Chapter 23.0, "System Management," for more details on the heartbeat packet format, and the specific bits sent in the packet.

26.5 Bus Arbitration

Several masters may attempt to get on the bus at the same time by driving the SMBDATA line low to signal a start condition. The Intel® 3100 Chipset continuously monitors the SMBDATA line. When the Intel® 3100 Chipset is attempting to drive the bus to a '1' by letting go of the SMBDATA line, and it samples SMBDATA low, then some other master is driving the bus and the Intel® 3100 Chipset stops transferring data.

If the Intel® 3100 Chipset detects loss of arbitration, the condition is called a collision. The Intel® 3100 Chipset sets the BUS_ERR bit in the Host Status register, and if enabled, generates an interrupt or SMI#. The processor is responsible for restarting the transaction.

26.6 Bus Timings

The SM Bus runs at between 10 – 100 kHz. Most of the timings associated with the SM Bus are microseconds in length. The SM Bus runs off of a divide by two of the RTC clock internally and employs counters of various length off of the RTC clock to drive the SM Bus.

Table 790. AC Timings on SM Bus

Timing	Min AC	Specification Name	RTC Clocks	AC Translation
t_{LOW}	4.7 µs	Clock low period	1	31.25 µs
t_{HIGH}	4.0 µs	Clock high period	1	31.25 µs
$t_{SU:DAT}$	250 ns	Data setup to rising SMBCLK	0.5	15.625 µs
$t_{HD:DAT}$	0 ns	Data hold from falling SMBCLK	0.5	15.625 µs
$t_{HD:STA}$	4.0 µs	Repeat Start Condition generated from rising SMBCLK	0.5	15.625 µs
$t_{SU:STA}$	4.7 µs	First clock fall from start condition	0.5	15.625 µs
$t_{SU:STO}$	4.0 µs	Last clock rising edge to last data rising edge (stop condition)	0.5	15.625 µs
t_{BUF}	4.7 µs	Time between consecutive transactions	1	31.25 µs

Note: The Minimum AC column indicates the minimum times required by the SMBus and/or I²C specifications. The Intel® 3100 Chipset tolerates these timings on both its SMLink and SMBus interfaces. For tHD_DAT, the minimum timing for I²C is 0 ns while the minimum timing for SMBus is 300 ns.

When the Intel® 3100 Chipset is a SM Bus master, it drives the clock. When the Intel® 3100 Chipset is sending address or command as an SM Bus master, or data bytes as a master on writes, it will drive data relative to the clock it is also driving. It does not start toggling the clock until the start or stop condition meets proper setup and hold. The Intel® 3100 Chipset also guarantees minimum time between SM Bus transactions as a master.

26.6.1 Clock Stretching

Some devices may not be able to handle their clock toggling at the rate that the Intel® 3100 Chipset as an SM Bus master would like. They have the capability of stretching the low time of the clock. When the Intel® 3100 Chipset attempts to release the clock (allowing the clock to go high), the clock remains low for an extended period of time.

The Intel® 3100 Chipset monitors the SM Bus clock line after it releases the bus to determine whether to enable the counter for the high time of the clock. While the bus is still low, the high time counter must not be enabled. Similarly, the low period of the clock can be stretched by an SM Bus master if it is not ready to send or receive data.

26.6.2 Bus Time Out (Intel® 3100 Chipset as SMB Master)

If there is an error in the transaction, such that an SMBus device does not signal an acknowledge, or holds the clock lower than the allowed time-out time, the transaction will time out. The Intel® 3100 Chipset discards the cycle, and set the DEV_ERR bit. The time out minimum is 25 ms. The time-out counter inside the Intel® 3100 Chipset starts when the first bit of data is transferred by the Intel® 3100 Chipset. The time-out minimum is 25 ms (800 RTC clocks).

The 25 ms timeout counter does not count under the following conditions:

1. BYTE_DONE_STATUS bit (SMBus I/O Offset 00h, bit 7) is set, and
2. The SECOND_TO_STS bit (TCO I/O Offset 06h, bit 1) is not set (this indicates that the system has not locked up).

26.7 Interrupts/SMI#

The Intel® 3100 Chipset SM Bus controller uses PIRQB# as its interrupt pin. However, the system can alternatively be set up to generate SMI# instead of an interrupt, by setting the SMBUS_SMI_EN bit (Device 31, Function 0, Offset 40h, bit 1).

The following tables specify how the various enable bits in the SMBus function control the generation of the interrupt, Host and Slave SMI, and Wake internal signals. The rows in the tables are additive, which means that if more than one row is true for a particular scenario then the results for all of the activated rows occurs.

Table 791. Summary of Enables for SMBALERT#

Event	INTREN (Host Control I/O Register, Offset 02h, Bit 0)	SMB_SMI_EN (Host Configuration Register, D31, F3, Offset 40h, Bit 1)	SMBALERT_DIS (Slave Command I/O Register, Offset 11h, Bit 2)	Result
SMBALERT# asserted low (always reported in SMBALERT_STS-Host Status Register, bit 5)	X	X	X	Wake generated
	X	1	0	Slave SMI# generated (SMBUS_SMI_STS)
	1	0	0	Interrupt generated

Table 792. Summary of Enables for SMBus Slave Write, and SMBus Host Events

Event	INTREN (Host Control I/O Register, Offset 02h, Bit 0)	SMB_SMI_EN (Host Configuration Register, D31, F3, Offset 40h, Bit 1)	Result
Slave Write to Wake/SMI# command	X	X	Wake generated when asleep Slave SMI# generated when awake (SMBUS_SMI_STS)
Slave Write to SMLINK_SLAVE_SMI command	X	X	Slave SMI# generated when in the S0 state (SMBUS_SMI_STS)
Any combination of Host Status Register [04:01] asserted	0	X	None
	1	0	Interrupt generated
	1	1	Host SMI# generated

Table 793. Summary of Enables for the Host Notify Command

HOST_NOTIFY_INTREN (Slave Control I/O Register, Offset 11h, bit 0)	SMB_SMI_EN (Host Configuration Register, D31, F3, Off40h, Bit 1)	HOST_NOTIFY_WKEN (Slave Control I/O Register, Offset 11h, bit 1)	Result
0	X	0	None
X	X	1	Wake generated
1	0	X	Interrupt generated
1	1	X	Slave SMI# generated (SMBUS_SMI_STS)

26.8 CRC Generation and Checking

If the AAC bit is set in the Auxiliary Control register, the Intel® 3100 Chipset automatically calculates and drives CRC at the end of the transmitted packet for write cycles, and checks the CRC for read cycles. It does not transmit the contents of the PEC register for CRC. The PEC bit must not be set in the Host Control register if this bit is set, or unspecified behavior results.

If the read cycle results in a CRC error, the DEV_ERR bit and the CRCE bit in the Auxiliary Status register at offset 0Ch is set.

26.9 Slave Interface I/O Space

The following registers are used by the SMBus Slave logic. Refer to Section 26.3 for the complete list of SMB I/O Registers.

Table 794. SMB Slave Interface I/O Registers Summary

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
09h	09h	RSA	Receive Slave Address Register	44h	RW
0Ah	0Bh	SD	Slave Data Register	0000h	RO
10h	10h	SSTS	Slave Status Register	00h	RWC
11h	11h	SCMD	Slave Command Register	00h	RW
14h	14h	NDA	Notify Device Address Register	00h	RO
16h	16h	NDLB	Notify Data Low Byte Register	00h	RO
17h	17h	NDHB	Notify Data High Byte Register	00h	RO

26.9.1 Register Details

26.9.1.1 Offset 09h: RSA – Receive Slave Address Register

Table 795. Offset 09h: RSA – Receive Slave Address Register

I/O Address: SMBBASE + 09h	*Size:* 8 bit			
Default Value: 44h	*Power Well:* Resume			
Bits	**Name**	**Description**	**Reset Value**	**Access**
07	Reserved	Reserved		
06:00	RSA	**SLAVE_ADDR[06:00]:** This field is the slave address that is decoded for read and write cycles. The default is not 0 so that it can respond even before the processor comes up (or if the processor is dead). This register is reset by CF9 RESET or RSMRST#, but not by PLTRST#.	1000100	RW

26.9.1.2 Offset 0Ah: SD – Slave Data Register

Table 796. Offset 0Ah: SD – Slave Data Register

I/O Address: SMBBASE + 0Ah	*Size:* 16 bit			
Default Value: 0000h	*Power Well:* Resume			
Bits	**Name**	**Description**	**Reset Value**	**Access**
15:00	SD	**SLAVE_DATA[15:00]:** This field is the 16-bit data value written by the external SMBus master. The processor can then read the value from this register. This register is reset by CF9 RESET or RSMRST#, but not by PLTRST#. SLAVE_DATA[07:00] corresponds to the Data Message Byte 0 (see Section 26.9.1.1) at Slave Write Register 4 in the table. SLAVE_[15:08] corresponds to the Data Message Byte 1 (see Section 26.9.1.1) at Slave Write Register 5 in the table.	0	RO

26.9.1.3 Offset 10h: SSTS – Slave Status Register

Table 797. Offset 10h: SSTS – Slave Status Register

I/O Address: SMBASE+10h			***Size:*** 8 bit	
Default Value: 00h			***Power Well:*** Resume (see Note below)	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:01	Reserved	Reserved	0	
00	HOST_ NOTIFY_ STS	Software reads this bit to determine that the source of the interrupt or SMI# was the reception of the Host Notify Command. Software clears this bit after reading any information needed from the notify address and data registers by writing a 1 to this bit. The Intel® 3100 Chipset allows the notify address and data registers to be overwritten once this bit has been cleared. When this bit is 1, the Intel® 3100 Chipset will NACK the first byte (host address) of any new "Host Notify" commands on the SMLink. Writing a 0 to this bit has no effect 0 = Bit is clear, allows the notify address and data registers to be overwritten. 1 = This bit is set when the Intel® 3100 Chipset has completely received a successful Host Notify Command on the SMLink pins.	0	RWC

Note: This register is in the resume well and is reset by CF9 RESET or RSMRST#. All bits in this register are implemented in the 64 kHz clock domain. Therefore, software must poll the register until a write takes effect before assuming that a write has completed internally.

26.9.1.4 Offset 11h: SCMD – Slave Command Register

Table 798. Offset 11h: SCMD – Slave Command Register

I/O Address: SMBASE+11h			***Size:*** 8 bit	
Default Value: 00h			***Power Well:*** Resume (see Note below)	
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:03	Reserved	Reserved	00	
02	SMBALERT_D IS	0 = Allows the generation of interrupt or SMI#. 1 = Software sets this bit to 1 to block the generation of the interrupt or SMI# due to the SMBALERT# source. This bit is logically inverted and ANDed with the SMBALERT_STS bit. The resulting signal is distributed to the SMI# and/or interrupt generation logic. This bit does not effect the wake logic.	0	RW
01	HOST_ NOTIFY_ WKEN	Software sets this bit to 1 to enable the reception of a Host Notify command as a wake event. When enabled this event is ORed with the other SMBus wake events and is reflected in the SMB_WAK_STS bit of the General Purpose Event 0 Status register. 0 = Disable 1 = Enable	0	RW
00	HOST_ NOTIFY_ INTREN	Software sets this bit to 1 to enable the generation of interrupt or SMI# when HOST_NOTIFY_STS is 1. This enable does not affect the setting of the HOST_NOTIFY_STS bit. When the interrupt is generated, either PIRQB or SMI# is generated, depending on the value of the SMB_SMI_EN bit (D31, F3, Off40h, B1). If the HOST_NOTIFY_STS bit is set when this bit is written to a 1, then the interrupt (or SMI#) will be generated. The interrupt (or SMI#) is logically generated by ANDing the STS and INTREN bits. 0 = Disable 1 = Enable	0	RW

Note: This register is in the resume well and is reset by CF9 RESET or RSMRST#. All bits in this register are implemented in a slow (64 khz) clock domain. Therefore, software must poll the register until a write takes effect before assuming that a write has completed internally. Also, software must confirm the prior written value before writing to the register again.

26.9.1.5 Offset 14h: NDA – Notify Device Address Register

This register is in the resume well and is reset by CF9 RESET or RSMRST#.

Table 799. Offset 14h: NDA – Notify Device Address Register

I/O Address: SMBASE+14h ***Size:*** 8 bit
Default Value: 00h ***Power Well:*** Resume; reset by CF9 RESET or RSMRST#

Bits	Name	Description	Reset Value	Access
07:01	DEVICE_ ADDRESS	This field contains the 7-bit device address received during the Host Notify protocol of the *SMBus Specification*. Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1.	0000000	RO
00	Reserved	Reserved	0	

26.9.1.6 Offset 16h: NDLB – Notify Data Low Byte Register

This register is in the resume well and is reset by CF9 RESET or RSMRST#.

Table 800. Offset 16h: NDLB – Notify Data Low Byte Register

I/O Address: SMBASE+16h ***Size:*** 8 bit
Default Value: 00h ***Power Well:*** Resume; reset by CF9 RESET or RSMRST#

Bits	Name	Description	Reset Value	Access
07:00	DATA_LOW_ BYTE	This field contains the first (low) byte of data received during the Host Notify protocol of the *SMBus Specification*. Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1.	00h	RO

26.9.1.7 Offset 17h: NDHB – Notify Data High Byte Register

This register is in the resume well and is reset by CF9 RESET or RSMRST#.

Table 801. Offset 17h: NDHB – Notify Data High Byte Register

I/O Address: SMBASE+17h ***Size:*** 8 bit
Default Value: 00h ***Power Well:*** Resume; reset by CF9 RESET or RSMRST#

Bits	Name	Description	Reset Value	Access
07:00	DATA_ HIGH_BYTE	This field contains the second (high) byte of data received during the Host Notify protocol of the *SMBus Specification*. Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1.	00h	RO

26.10 Slave Interface Behavioral Description

the The SMBus slave logic does not generate or handle receiving the PEC byte. The SMBus slave logic only acts as a “Legacy ASF” device; no modifications are made for Internal ASF operation.

The slave interface allows the Intel® 3100 Chipset to decode cycles on SMLink in TCO compatible mode, and allows an external microcontroller to perform specific actions. Key features and capabilities:

- Supports decode of three types of messages: byte write, byte read, and host notify
- Register for the receive slave address. This is the address that the Intel® 3100 Chipset decodes. A default value is provided so that the slave interface can be used without the processor having to program this register.
- Receive Slave Data register in the SMBus I/O space that includes the data written by the external microcontroller
- Registers that the external microcontroller can read to get the state
- Status bits to indicate that the SMLink/SMBus slave logic caused an interrupt or SMI#
  - — Bit 0 of the slave status register for the host notify command
  - — Bit 16 of the SMI Status Register for all others

Note: The external microcontroller should not attempt to access the SMBus slave logic until one second after both: RTEST# is high and RSMRST# is high.

If a master leaves the clock and data bits of the SMLink or SMBus interface at '1' for 50 µs or more in the middle of a cycle, the slave logic's behavior is undefined. This is interpreted as an unexpected idle and should be avoided when performing management activities to the slave logic.

26.10.1 Format of Slave Write Cycle

The external master performs byte write commands to the SMBus Slave Interface. The Command field (bits 11 – 18) indicate which register is being accessed. The Data field (bits 20 – 27) indicate the value that should be written to that register.

The write cycle format is shown below in Table 802. Table 803 has the values associated with the registers.

Table 802. Slave Write Cycle Format

Bit	Description	Driven By	Comment
1	Start Condition	External Microcontroller	
2 – 8	Slave Address - 7 bits	External Microcontroller	Must match value in Receive Slave Address register
9	Write	External Microcontroller	Hardwired to 0
10	ACK	Intel® 3100 Chipset	
11 – 18	Command	External Microcontroller	This field indicates which register will be accessed. See Table 803 for the register definitions
19	ACK	Intel® 3100 Chipset	
20 – 27	Register Data	External Microcontroller	See Table 803 for the register definitions
28	ACK	Intel® 3100 Chipset	
29	Stop	External Microcontroller	

Table 803. Slave Write Registers

Register	Function:
0	Command Register. See Table 804 for legal values written to this register.
1 – 3	Reserved
4	Data Message Byte 0
5	Data Message Byte 1
6 – 7	Reserved
8	Reserved
9 – FFh	Reserved

Note: The external microcontroller is responsible to make sure that it does not update the contents of the data byte registers until they have been read by the system processor. The Intel® 3100 Chipset overwrites the old value with any new value received. A race condition is possible where the new value is being written to the register just at the time it is being read. The Intel® 3100 Chipset does not attempt to cover this race condition (i.e., unpredictable results in this case).

Table 804. Command Types

Command Type	Description
0	Reserved
1	**WAKE/SMI#:** Wake system if it is not already awake. If system is already awake, then an SMI# is generated.
2	**Unconditional Powerdown:** This command should set the PWRBTNOR_STS bit, and have the same effect as the power button override occurring.
3	**HARD RESET Without Power Cycling SYSTEM:** The causes a soft reset of the system (does not include cycling of the power supply). This is equivalent to a write to the CF9h register with bits 02:01 set to 1, but bit 03 set to 0.
4	**HARD RESET SYSTEM:** The causes a hard reset of the system (including cycling of the power supply). This is equivalent to a write to the CF9h register with bits 03:01 set to 1.
5	**Disable the TCO Messages.** This command disables the IICH from sending Heartbeat and Event messages. Once this command has been done, there is no method to reenable the Heartbeat and Event messages, until CF9 RESET or RSMRST# goes low and then high.
6	**WD RELOAD:** Reload watchdog timer.
7	Reserved
8	**SMLINK_SLAVE_SMI:** When the Intel® 3100 Chipset detects this command type while in the S0 state, it will set the SMLINK_SLAVE_SMI_STS bit. This command should only be used if the system is in an S0 state. If the message is received during S3 or S5 states, it is acknowledged by the Intel® 3100 Chipset but the SMLINK_SLAVE_SMI_STS bit is not set. ***Note:*** It is possible that the system transitions out of the S0 state at the same time that the SMLINK_SLAVE_SMI command is received. In this case, the SMLINK_SLAVE_SMI_STS bit may get set but not serviced before the system goes to sleep. Once the system returns to S0, the SMI associated with this bit would then be generated. Software must be able to handle this scenario.
9 – FFh	Reserved

26.10.2 Format of Read Command

The external master performs byte read commands to the SMBus Slave Interface. The Command field (bits 11-18) indicate which register is being accessed. The Data field (bits 30-37) indicate the value that should be read from that register. Table 805 shows the read cycle format. Table 806 shows the register mapping for the data byte.

Table 805. Slave Read Cycle Format

Bit	Description	Driven by:	Comment:
1	Start	External Microcontroller	
2 – 8	Slave Address - 7 bits	External Microcontroller	Must match value in Receive Slave Address register
9	Write	External Microcontroller	Hardwired to 0
10	ACK	Intel® 3100 Chipset	
11 – 18	Command code - 8 bits	External Microcontroller	Indicates which register is being accessed. See Table 806 for list of implemented registers.
19	ACK	Intel® 3100 Chipset	
20	Repeated Start	External Microcontroller	
21 – 27	Slave Address - 7 bits	External Microcontroller	Must match value in Receive Slave Address register
28	Read	External Microcontroller	Hardwired to 1
29	ACK	Intel® 3100 Chipset	
30 – 37	Data Byte	Intel® 3100 Chipset	Value depends on register being accessed. See Table 806 for list of implemented registers.
38	NOT ACK	External Microcontroller	
39	Stop	External Microcontroller	

Table 806. Data Values for Slave Read Registers (Sheet 1 of 2)

Register	Bits	Description
0	07:00	Reserved for capabilities indication. Should always return 00h. Future chips may return another value to indicate different capabilities.
1	02:00	System Power State 000 = S0 001 = Reserved 010 = Reserved 011 = S3 100 = Reserved 101 = S5 110 = Reserved 111 = Reserved
	07:03	Reserved
2	03:00	Reserved
	07:04	Reserved
3	05:00	Watchdog Timer current value. Watchdog Timer has 10 bits, but this field is only 6 bits. If the current value is greater than 3Fh, Intel® 3100 Chipset always reports 3Fh in this field.
	07:06	Reserved

Table 806. Data Values for Slave Read Registers (Sheet 2 of 2)

Register	Bits	Description
4	00	1 = The Intruder Detect (INTRD_DET) bit is set. This indicates that the system cover has been opened.
	01	1 = BTI Temperature Event occurred. This bit is set if Intel® 3100 Chipset's THRM# input signal is active. Need to take after polarity control.
	02	DOA processor Status. This bit is 1 to indicate that the processor is dead.
	03	1 = SECOND_TO_STS bit set. This bit is set after the second timeout (SECOND_TO_STS bit) of the Watchdog Timer occurs.
	06:04	Reserved. Will always be 0, but software should ignore.
	07	Reflects the value of the GPI[11]/SMBALERT# pin (and is dependent upon the value of the GPI_INV[11] bit. If the GPI_INV[11] bit is 1, then the value in this bit equals the level of the GPI[11]/SMBALERT# pin (high = 1, low = 0). If the GPI_INV[11] bit is 0, then the value of this bit equals the inverse of the level of the GPI[11]/SMBALERT# pin (high = 0, low = 1).
5	00	FWH bad bit. This bit is 1 to indicate that the FWH read returned FFh, which indicates that it is probably blank.
	01	Battery Low Status. '1' if the BATLOW# pin is a '0'.
	02	CPU Power Failure Status: '1' if the CPUPWR_FLR bit in the GEN_PMCON_2 register is set.
	07:01	Reserved
6	07:00	Contents of the Message 1 register. See Section 23.2.2.8 for details.
7	07:00	Contents of the Message 2 register. See Section 23.2.2.8 for details.
8	07:00	Contents of the WDSTATUS register. See Section 23.2.2.9 for details.
9	07:00	Reserved
A	07:00	Reserved
B	07:00	Reserved
C – FFh	07:00	Reserved

Warning: The external microcontroller is responsible to make sure that it does not read the contents of the various message registers until they have been written by the system processor. Intel® 3100 Chipset overwrites the old value with any new value received. A race condition is possible where the new value is being written to the register just at the time it is being read. Intel® 3100 Chipset does not attempt to cover this race condition (i.e., unpredictable results in this case).

Behavioral Notes:

The SMBus protocol always has either start bit-address-write bit or repeated start bit-address-read bit. Intel® 3100 Chipset is implemented such that the read/write bit in the repeated start phase is ignored with an assumption that the protocol always followed. In other words, if start-address-read occurs (which is illegal for SMBus byte read protocol), Intel® 3100 Chipset still grabs the cycle. In another case, if a repeated start-address-write sequence occurs, then the cycle continues as a slave read.

26.10.3 Format of the Host Notify Command

Intel® 3100 Chipset tracks and responds to the standard Host Notify command as specified in the *SMBus Specification*. The host address for this command is fixed to 0001000b. If Intel® 3100 Chipset already has data for a previously-received host notify command which has not been serviced yet by the host software (as indicated by the

HOST_NOTIFY_STS bit), then it will NACK following the host address byte of the protocol. This allows the host to communicate non-acceptance to the master and retain the host notify address and data values for the previous cycle until host software completely services the interrupt.

Note: Host software must always clear the HOST_NOTIFY_STS bit *after* completing any necessary reads of the address and data registers.

Table 807 shows the host notify protocol.

Table 807. Host Notify Protocol

Bit	Description	Driven by:	Comment:
1	Start	External Master	
2 – 8	SMB Host Addr - 7 bits	External Master	Always 0001_000
9	Write	External Master	Hardwired to 0
10	ACK (or NACK)	Intel® 3100 Chipset	NACKs if HOST_NOTIFY_STS is 1
11 – 17	Device Address - 7 bits	External Master	Indicates the address of the master; loaded in to the Notify Device Address Register
18	Unused- Hardwired to 0	External Master	7-bit-only address; this bit is inserted to complete the byte
19	ACK	Intel® 3100 Chipset	
20 – 27	Data Byte Low	External Master	Loaded in to the Notify Data Low Byte Register
28	ACK	Intel® 3100 Chipset	
29 – 36	Data Byte High	External Master	Loaded in to the Notify Data High Byte Register
37	ACK	Intel® 3100 Chipset	
38	Stop	External Master	

27.0 High Precision Event Timers (HPET)

Note: The name "Multimedia Timers (MMT)" has been replaced with "High-Precision Event Timers (HPET)". This section documents the Intel® 3100 Chipset-specific behavior and the generic *HPET Specification*.

27.1 Overview

This function provides a set of timers that can be used by the operating system. The timers are defined such that in the future, the OS may be able to assign specific timers to be used directly by specific applications. Each timer can be configured to cause a separate interrupt. This specification allows for a block of 32 timers, with support for up to eight blocks, for a total of 256 timers. The timers are implemented as a single counter with a set of comparators. The counter increases monotonically. Each timer includes a value register and a comparator. Each individual timer can generate an interrupt when the value in its value register matches value in the main counter. Some of the timers can be enabled to generate a periodic interrupt.

The registers associated with these timers are mapped to a memory space (much like the I/O APIC). However, it is not implemented as a standard PCI function. The BIOS reports to the operating system the location of the register space. The hardware can support an assignable decode space; However, the BIOS sets this space prior to handing it over to the OS. It is not expected that the OS move the location of these timers once they are set by the BIOS.

27.2 Registers

The timer registers are memory mapped in a non-indexed scheme. This allows the processor to directly access each register without having to use an index register. The timer register space is 1024 bytes. The registers are generally aligned on 64-bit boundaries to simplify implementation with IA-64 processors.

General Behavioral Rules:

- Software must not attempt to read or write across register boundaries. For example, a 32-bit access must be to offset x0h, x4h, x8h, or xCh.
- 32-bit accesses must not be to 01h, 02h, 03h, 05h, 06h, 07h, 09h, 0Ah, 0Bh, 0Dh, 0Eh, or 0Fh. Any accesses to these offsets results in an unexpected behavior and may result in a master abort. However, these accesses may not result in system hangs.
- 64-bit accesses can only be to x0h and must not cross 64-bit boundaries.
- Software must not write to read-only registers.
- Software must not expect any particular or consistent value when reading reserved registers or bits.
- The timer register space is memory mapped to a 1 K block.
- There are four possible memory address ranges beginning at:
  - FED0_0000h

— FED0_1000h

— FED0_2000h

— FED0_3000h

- The choice of address ranges will be selected by configuration bits in the High Performance Timer Configuration Register (in the memory mapped-configuration area).
- All registers are implemented in the Core well, and all bits are reset by PLTRST#.
- Reads to reserved registers or bits return a value of 0.

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 808. High Precision Event Timers Registers Summary

Offset		Symbol	Register Name/Function	Default	Type
Start	End				
000h	007h	GCAP_ID	General Capabilities and ID Register	0429B17F 8086A201h	RO
010h	017h	GEN_CONF	General Configuration Register	00000000_ 00000000h	RW
020h	027h	GINTR_STA	General Interrupt Status Register	00000000_ 00000000h	RWC
0F0h	0F7h	MAIN_CNT	Main Counter Value Register	Xh	RW
100h	107h	TIM1_CONF	Timer 0 Configuration and Capabilities Register	Xh	RW
108h	10Fh	TIM1_COMP	Timer 0 Comparator Value Register	Xh	RW
120h	127h	TIM2_CONF	Timer 1 Configuration and Capabilities Register	Xh	RW
128h	12Fh	TIM2_COMP	Timer 1 Comparator Value Register	Xh	RW
140h	147h	TIM3_CONF	Timer 2 Configuration and Capabilities Register	Xh	RW
148h	14Fh	TIM3_COMP	Timer 2 Comparator Value Register	Xh	RW

Notes:
1. Reads to reserved registers or bits returns a value of 0.
2. Software must not attempt to lock the memory-mapped I/O ranges for High-Precision Timers. If attempted, the lock is not honored, which means potential deadlock conditions may occur.

27.2.1 Register Details

27.2.1.1 Offset 000 - 007h: GCAP_ID - General Capabilities and ID Register

General Behavioral Rules:

- Writes to this register must not be attempted by software.
- Software can read the various bytes in this register using 32-bit or 64-bit accesses.
- 32-bit accesses can be done to offset 00h or 04h, but not to offsets 01h, 02h, 03h, 05h, 06h, or 07h.
- 64-bit accesses can only be done to 00h.

Table 809. Offset 000 - 007h: GCAP_ID - General Capabilities and ID Register

I/O Address: 000 - 007h ***Size:*** 64 bit

Default Value: 0429B17F8086A201h

Bits	Name	Description	Reset Value	Access
63:32	COUNTER_ CLK_PER_CAP	**Main Counter Tick Period:** This read-only field indicates the period at which the counter increments in femptoseconds (10^-15 seconds). This returns 0429B17Fh when read indicating a period of 69841279 fs (69.841279 ns).	0429B17Fh	RO
31:16	VENDOR_ID_ CAP	**Vendor ID Capability:** These bits return 8086h when read. This is a 16-bit value assigned to Intel.	8086h	RO
15	LEG_RT_CAP	**Legacy Replacement Rout Capable:** This bit is always one when read, as the Legacy Replacement Interrupt Rout is supported.	1	RO
14	Reserved	**Reserved:** This bit returns zero when read.	0	
13	COUNT_SIZE_C AP	**Counter Size Capability:** Indicates that the main counter is 64 bits wide. This bit returns one when read.	1	RO
12:08	NUM_TIM_CAP	**Number of Timer Capability:** This field indicates the number of timers in this block. This value in this field is 02h = Three timers.	02h	RO
07:00	REV_ID	**Revision Identification:** This field indicates which revision of the specification is implemented. The value in this field is 01h.	01h	RO

27.2.1.2 Offset 010 - 017h: GEN_CONF - General Configuration Register

General Behavioral Rules:

- Software can access the various bytes in this register using 32-bit or 64-bit accesses.
- 32-bit accesses can be done to offset 010h or 014h, but not to offsets 011h, 012h, 013h, 015h, 016h, or 017h.
- 64-bit accesses can only be done to 010h.

Table 810. Offset 010 - 017h: GEN_CONF - General Configuration Register

I/O Address: 010 - 017h ***Size:*** 64 bit

Default Value: *00000000_00000000h*

Bits	Name	Description	Reset Value	Access
63:02	Reserved	**Reserved:**	000h	
01	LEG_RT_CNF	**Legacy Replacement Route:** If the ENABLE_CNF and LEG_RT_CNF bits are set, then the interrupts are routed as follows: Timer 0 is routed to IRQ0 in 8259 or IRQ2 in the I/O APIC Timer 1 is routed to IRQ8 in 8259 or IRQ8 in the I/O APIC Timer 2-n is routed as per the routing in the timer *n* config registers. 0 = If the LEG_RT_CNF bit is not set, the individual routing bits for each of the timers are used. 1 = Legacy Rout: If the LEG_RT_CNF bit is set, the individual routing bits for timers 0 and 1 (APIC or FSB) have no impact.	0b	RW
00	ENABLE_CNF	**Overall Enable:** 0 = The main counter halts (does not increment) and no interrupts are caused by any of these timers. 1 = Enable any of the timers to generate interrupts. For level-triggered interrupts, if an interrupt is pending when the ENABLE_CNF bit is changed from 1 to 0, the interrupt status indications (in the various Txx_INT_STS bits) are not cleared. Software must write to the Txx_INT_STS bits to clear the interrupts.	0b	RW

27.2.1.3 Offset 020 - 027h: GINTR_STA - General Interrupt Status Register

General Behavioral Rules:

- Software can access the various bytes in this register using 32-bit or 64-bit accesses.
- 32-bit accesses can be done to offset 10h or 14h, but not to offsets 11h, 12h, 13h, 15h, 16h, or 17h.
- 64-bit accesses can only be done to 10h.

Table 811. Offset 020 - 027h: GINTR_STA - General Interrupt Status Register

I/O Address: 020 - 027h ***Size:*** 64 bit

Default Value: 00000000_00000000h

Bits	Name	Description	Reset Value	Access
63:03	Reserved	**Reserved**	00h	
02	T02_INT_STS	**Timer 2 Interrupt Active:** Same functionality as Timer 0.	0	RW
01	T01_INT_STS	**Timer 1 Interrupt Active:** Same functionality as Timer 0.	0	RW
00	T00_INT_STS	**Timer 0 Interrupt Active:** The functionality of this bit depends on whether the edge or level-triggered mode is used for this timer: If set to level-triggered mode: This bit is set by hardware if the corresponding timer interrupt is active. Once the bit is set, it can be cleared by software writing a 1 to the same bit position. Writes of 0 to this bit have no effect. If set to edge-triggered mode: • This bit must be ignored by software. Software must always write 0 to this bit.	0	RWC

27.2.1.4 Offset 0F0 - 0F7h: MAIN_CNT - Main Counter Value Register

General Behavioral Rules:

- Software can access the various bytes in this register using 32-bit or 64-bit accesses.
- 32-bit accesses can be done to offset 0F0h or 0F4h.
- 32-bit accesses must not be done starting at: 0F1h, 0F2h, 0F3h, 0F5h, 0F6h, or 0F7h.
- 64-bit accesses can be done to 0F0h.
- Writes to this register must only be done while the counter is halted.
- Reads to this register return the current value of the main counter.
- 32-bit counters will always return zero for the upper 32 bits of this register.
- If 32-bit software attempts to read a 64-bit counter, it must first halt the counter. Since this will delay the interrupts for all of the timers, this must be done only if the consequences are understood. It is strongly recommended that 32-bit software only operate the timer in 32-bit mode.

Table 812. Offset 0F0 - 0F7h: MAIN_CNT - Main Counter Value Register

I/O Address: 0F0 - 0F7h ***Size:*** 64 bit
Default Value: *Xh*

Bits	Name	Description	Reset Value	Access
63:00	COUNTER_VAL[63:00]	**Counter Value:** Bits 63:00 of the counter. ***Notes:*** 1. Writes to this register must only be done while the counter is halted. 2. Reads to this register return the current value of the main counter. 3. 32-bit counters always return zero for the upper 32 bits of this register. 4. If 32-bit software attempts to read a 64-bit counter, it must first halt the counter. Since this delays the interrupts for all of the timers, this must be done only if the consequences are understood. It is strongly recommended that 32-bit software only operates the timer in 32-bit mode. 5. Reads to this register are monotonic. No two consecutive reads return the same value. The second of two reads always returns a larger value, unless the timer has rolled over to 0.	Xh	RW

27.2.1.5 Timer *n* Configuration and Capabilities Register

General Behavioral Rules:

- Software can access the various bytes in this register using 32-bit or 64-bit accesses.
- 32-bit accesses can be done to offset 1x0h or 1x4h. 64-bit accesses can be done to 1x0h.
- 32-bit accesses must not be done to 1x1h, 1x2h, 1x3h, 1x5h, 1x6h, 1x7h.

Note: The letter *n* can be 0, 1 or 2, referring to Timer 0, 1 or 2.

Table 813. Timer *n* Configuration and Capabilities Register (Sheet 1 of 2)

I/O Address:	Timer 0: 100 – 107h, Timer 1: 120 – 127h, Timer 2: 140 – 147h, Timer *n*: (20h * *n*) +100h - (20h * *n*) + 107h		***Size:*** 64 bit	
Default Value: *Xh*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
63:56	Reserved	**Reserved:** These bits return 0 when read.	0h	
55:52, 43	TIMERn_INT_ROUT_CAP	**Timer Interrupt Route Capability:** Timer 0, 1:Bits 52, 53, 54, and 55 in this field (corresponding to IRQ 20, 21, 22, and 23) have a value of 1. Writes will have no effect. Timer 2:Bits 43, 52, 53, 54, and 55 in this field (corresponding to IRQ 11, 20, 21, 22, and 23) have a value of 1. Writes will have no effect. ***Note:*** If IRQ 11 is used for High Precision Event Timer #2, software must ensure IRQ 11 is not shared with any other devices to guarantee the proper operation of High Precision Event Timer #2.	X	RO
51:44, 42:14	Reserved	**Reserved:** These bits return 0 when read.	0	
13:09	TIMERn_INT_ROUT_CNF	**Interrupt Route:** (where *n* is the timer number: 00 to 31). This 5-bit field indicates the routing for the interrupt to the I/O APIC. A maximum value of 32 interrupts is supported. The default is 00h. Software writes to this field to select which interrupt in the I/O (X)APIC used for this timer's interrupt. If the value is not supported by this particular timer, then the value read back does not match what is written. The software must only write valid values. ***Notes:*** 1. If the Legacy Replacement Rout bit is set, then Timers 0 and 1 have a different routing, and this bit field has no effect for those two timers. 2. Timer 0,1: The software is responsible to make sure it programs a valid value (decimal 20, 21, 22, or 23) for this field. The logic does not check the validity of the value written. 3. Timer 2: The software is responsible to make sure it programs a valid value (decimal 11, 20, 21, 22, or 23) for this field. The logic does not check the validity of the value written.	X	RW
08	TIMERn_32MODE_CNF	**Timer *n* 32-bit Mode:** (where *n* is the timer number: 00 to 31). Software can set this bit to force a 64-bit timer to behave as a 32-bit timer. This is typically needed if the software is not willing to halt the main counter to read or write a particular timer. This bit is only relevant if the timer is operating in 64-bit mode in which case that timer can be forced to 32-bit mode by setting this bit. When Timer 0 is switched to 32-bit mode, the upper 32 bits are loaded with all 0's which will remain when the timer is switched back to 64-bit mode. If the timer is not in 64-bit mode, then this bit will always be read as 0 and writes will have no effect. Timer 0: Read/Write (default 0). 0 = 64-bit, 1 = 32-bit Timers 1/2: Writes have no effect since these timers are 32-bit only	X	RW or RO
07	Reserved	**Reserved:**	0b	

Note: Reads or writes to unimplemented timers must not be attempted. Reads from any unimplemented registers return an undetermined value.

Table 813. Timer *n* Configuration and Capabilities Register (Sheet 2 of 2)

I/O Address:	Timer 0: 100 – 107h, Timer 1: 120 – 127h, Timer 2: 140 – 147h, Timer *n*: (20h * *n*) +100h - (20h * *n*) + 107h		***Size:*** 64 bit	
Default Value: *Xh*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
06	TIMERn_VAL_SET_CNF	**Timer n Value Set:** Software uses this bit only for timers that have been set to periodic mode. 0 = Disabled. Software does NOT have to write this bit back to 0 (it automatically clears). 1 = By writing this bit to a 1, the software is allowed to directly set the timer's accumulator. ***Note:*** Software must not write a 1 to this bit position if the timer is set to non-periodic mode. ***Note:*** This bit returns zero when read. Writes will only have an effect for Timer 0 if it is set to periodic mode. Writes have no effect for Timers 1 and 2.	X	RW
05	TIMERn_SIZE_CAP	**Timer *n* Size:** (where *n* is the timer number: 00 to 31). This read-only field indicates the size of the timer. 0 = 32 bits 1 = 64 bits Timer 0: Value is 1 (64 bits). Timers 1 and 2: Value is 0 (32 bits).	X	RO
04	TIMERn_PER_INT_CAP	**Periodic Interrupt Capable:** (where *n* is the timer number: 00 to 31). If this read-only bit is 1, then the hardware supports a periodic mode for this timer's interrupt. Timer 0: Hardwired to 1 (supports the periodic interrupt). Timers 1 and 2: Hardwired to 0 (does not support periodic interrupts), so the bit is always read as zero.	X	RO
03	TIMERn_TYPE_CNF	**Timer n Type:** (where *n* is the timer number: 00 to 31). Timer 0:Bit is read/write. 0 = Disable timer to generate a periodic interrupt. 1 = Enable timer to generate a periodic interrupt. Timers 1, 2: Hardwired to 0.	X	RW
02	TIMERn_INT_ENB_CNF	**Timer n Interrupt Enable:** (where *n* is the timer number: 00 to 31). This bit must be set to enable timer n to cause an interrupt when it times out. 0 = Disable. The timer still counts and generates appropriate status bits, but does not cause an interrupt. 1 = Enable.	0	RW
01	TIMERn_INT_TYPE_CNF	**Timer Interrupt Type:** (where *n* is the timer number: 00 to 31) 0 = The timer interrupt is edge triggered. This means that an edge-type interrupt is generated. If another interrupt occurs, another edge is generated. 1 = The timer interrupt is level triggered. This means that a level-triggered interrupt is generated. The interrupt is held active until it is cleared by writing to the bit in the General Interrupt Status Register. If another interrupt occurs before the interrupt is cleared, the interrupt remains active.	0	RW
00	Reserved	**Reserved:** This bit returns zero when read.	0	

Note: Reads or writes to unimplemented timers must not be attempted. Reads from any unimplemented registers return an undetermined value.

27.2.1.6 Timer *n* Comparator Value Register

General Behavioral Rules:

- Software can access the various bytes in this register using 32-bit or 64-bit accesses.

- 32-bit accesses can be done to offset 1x8h or 1xCh. 64-bit accesses can be done to 1x8h.
- 32-bit accesses must not be done to 1x9h, 1xAh, 1xBh, 1xDh, 1xEh, or 1xFh.
- Reads to this register return the current value of the comparator.
- If the timer is configured to non-periodic mode:
  - Writes to this register load the value against which the main counter must be compared for this timer.
  - When the main counter equals the value last written to this register, the corresponding interrupt can be generated (if so enabled).
  - The value in this register does not change based on the interrupt being generated.
- If the timer is configured to periodic mode:
  - When the main counter equals the value last written to this register, the corresponding interrupt can be generated (if so enabled).
  - After the main counter equals the value in this register, the value in this register is increased by the value last written to the register.
- For example, if the value written to the register is 00000123h, then:
  - An interrupt is generated when the main counter reaches 00000123h.
  - The value in this register is then adjusted by the hardware to 00000246h.
  - Another interrupt is generated when the main counter reaches 00000246h.
  - The value in this register is then adjusted by the hardware to 00000369h.
- As each periodic interrupt occurs, the value in this register increments. When the incremented value is greater than the maximum value possible for this register (FFFFFFFFh for a 32-bit timer or FFFFFFFFFFFFFFFFh for a 64-bit timer), the value wraps around through 0. For example, if the current value in a 32-bit timer is FFFF0000h and the last value written to this register is 20000, then after the next interrupt the value changes to 00010000h.
- Default value for each timer is all 1's for the bits that are implemented. For example, a 32-bit timer has a default value of 00000000FFFFFFFFh. A 64-bit timer has a default value of FFFFFFFFFFFFFFFFh.

27.3 Theory Of Operation

Table 814. Timer n Comparator Value Register

I/O Address:	Timer 0: 108 - 10Fh Timer 1: 128 - 12Fh Timer 2: 148 - 14Fh	***Size:*** 64 bit	
Default Value: *Xh*			

Bits	Name	Description	Reset Value	Access
63:00	TIMn_COMP	**Timer Compare Value:** Reads to this register return the current value of the comparator. Timers 0, 1, or 2 are configured to non-periodic mode: Writes to this register load the value against which the main counter must be compared for this timer. • When the main counter equals the value last written to this register, the corresponding interrupt can be generated (if so enabled). • The value in this register does not change based on the interrupt being generated. Timer 0 is configured to periodic mode: • When the main counter equals the value last written to this register, the corresponding interrupt can be generated (if so enabled). • After the main counter equals the value in this register, the value in this register is increased by the value last written to the register. • For example, if the value written to the register is 00000123h, then: 1. An interrupt is generated when the main counter reaches 00000123h. 2. The value in this register is then adjusted by the hardware to 00000246h. 3. Another interrupt is generated when the main counter reaches 00000246h. 4. The value in this register is then adjusted by the hardware to 00000369h. • As each periodic interrupt occurs, the value in this register increments. When the incremented value is greater than the maximum value possible for this register (FFFFFFFFh for a 32-bit timer or FFFFFFFFFFFFFFFFh for a 64-bit timer), the value wraps around through zero. For example, if the current value in a 32-bit timer is FFFF0000h and the last value written to this register is 20000, then after the next interrupt the value changes to 00010000h. Default value for each timer is all 1s for the bits that are implemented. For example, a 32-bit timer has a default value of 00000000FFFFFFFFh. A 64-bit timer has a default value of FFFFFFFFFFFFFFFFh.	Xh	R/W

27.3.1 Timer Accuracy Rules

- The timers are expected to be accurate over any 1 ms period to within 0.05% of the time specified in the timer resolution fields.
- Within any 100 µs period, the timer is permitted to report a time that is up to two ticks too early or too late. Each tick must be less than or equal to 100 ns; this represents an error of less than 0.2%.
- The timer must be monotonic. It must never return the same value on two consecutive reads (unless the counter has rolled over and actually reached the same value).

- The main counter is clocked by the 14.31818 MHz clock, synchronized into the 66.666 MHz domain. This results in a non-uniform duty cycle on the synchronized clock, but does have the correct average period. The accuracy of the main counter is as accurate as the 14.3818 MHz clock.

27.3.2 Interrupt Mapping

The interrupts associated with the various timers have several interrupt mapping options. When reprogramming the HPET (High Precision Event Timer) interrupt routing scheme (LEG_RT_CNF bit in the General Config Register), a spurious interrupt may occur. This is because the other source of the interrupt (8254 timer) may be asserted. Software must mask interrupts prior to clearing the LEG_RT_CNF bit.

Mapping Option 1: Legacy Replacement Option

In this case, the Legacy Rout bit (LEG_RT_CNF) is set. This forces the mapping found in Table 815.

Table 815. Legacy Replacement Routing

Timer	8259 Mapping	APIC Mapping	Comment
0	IRQ0	IRQ2	In this case, the 8254 timer does not cause any interrupts
1	IRQ8	IRQ8	In this case, the RTC does not cause any interrupts.
2	As per IRQ Routing Field	As per IRQ Routing Field	

Mapping Option 2: Standard Option

In this case, the Legacy Rout bit (LEG_RT_CNF) is zero. Each timer has its own routing control. The interrupts can be routed to various interrupts in the I/O APIC. A capabilities field indicates which interrupts are valid options for the routing.

If a timer is set for edge-triggered mode, the timers must not be shared with any PCI interrupts.

Supported interrupt values are IRQ 20, 21, 22, and 23.

27.3.3 Periodic vs. Non-Periodic Modes

27.3.3.1 Non-Periodic Mode

This mode can be thought of as creating a one-shot.

Timer 0 is configurable to 32-bit (default) or 64-bit mode, whereas Timers 1 and 2 only support 32-bit mode (See Table 813).

All three timers support non-periodic mode.

When a timer is set up for non-periodic mode, it generates a value in the main counter that matches the value in the timer's comparator register. If the timer is set up for 32-bit mode, then it generates another interrupt when the main counter wraps around.

During run-time, the value in the timer's comparator value register is not changed by the hardware. Software can, of course, change the value.

The Timer 0 Comparator Value register cannot be programmed reliably by a single 64-bit write in a 32-bit environment **except** if only the periodic rate is being changed during run-time. If the actual Timer 0 Comparator Value needs to be reinitialized, then the following software solution always works regardless of the environment:

1. Set TIMER0_VAL_SET_CNF bit.
2. Set the lower 32 bits of the Timer0 Comparator Value register.
3. Set TIMER0_VAL_SET_CNF bit.
4. Set the upper 32 bits of the Timer0 Comparator Value register.

Warning: Software must be careful when programming the comparator registers. If the value written to the register is not sufficiently far in the future, then the counter may pass the value before it reaches the register and the interrupt will be missed. The BIOS will pass a data structure to the OS to indicate that the OS must not attempt to program the periodic timer to a rate faster then X. For the Intel® 3100 Chipset, X is 5 microseconds.

Every timer is required to support the non-periodic mode of operation.

27.3.3.2 Periodic Mode

When a timer is set up for periodic mode, the software writes a value in the timer's comparator value register. When the main counter value matches the value in the timer's comparator value register, an interrupt is generated. The hardware then automatically increases the value in the comparator value register by the last value written to that register.

To make the periodic mode work properly, the main counter is typically written with a value of 0 so that the first interrupt occurs at the right point for the comparator. If the main counter is not set to 0, interrupts may not occur as expected.

During run-time, the value in the timer's comparator value register can be read by software to find out when the next periodic interrupt will be generated (not the rate at which it generates interrupts). Software is expected to remember the last value written to the comparator's value register (the rate at which interrupts are generated).

If software wants to change the periodic rate, it must write a new value to the comparator value register. At the point when the timer's comparator indicates a match, this new value is added to derive the next matching point.

If the software resets the main counter, the value in the comparator's value register needs to reset as well. This can be done by setting the TIMERn_VAL_SET_CNF bit. Again, to avoid race conditions, this must be done with the main counter halted. The following usage model is expected:

Warning: As the timer period approaches zero, the interrupts associated with the periodic timer may not get completely serviced before the next timer match occurs. Interrupts may get lost and/or system performance may be degraded in this case.

Each timer is NOT required to support the periodic mode of operation. A capabilities bit indicates if the particular timer supports periodic mode. The reason for this is that supporting the periodic mode adds a significant number of gates.

For the Intel® 3100 Chipset, only Timer 0 supports periodic mode. The following usage model is expected:

1. Software clears the ENABLE_CNF bit to prevent any interrupts.
2. Software clears the main counter by writing a value of 00h to it.
3. Software sets the TIMER0_VAL_SET_CNF bit.

4. Software writes the new value in the TIMER0_COMPARATOR_VAL register.
5. Software sets the ENABLE_CNF bit to enable interrupts.

The Timer 0 Comparator Value register cannot be programmed reliably by a single 64-bit write in a 32-bit environment except if only the periodic rate is being changed during run-time. If the actual Timer 0 Comparator Value needs to be reinitialized, then the following software solution will always work regardless of the environment:

1. Set TIMER0_VAL_SET_CNF bit
2. Set the lower 32 bits of the Timer0 Comparator Value register
3. Set TIMER0_VAL_SET_CNF bit
4. Set the upper 32 bits of the Timer0 Comparator Value register

27.3.4 Enabling the Timers

The BIOS or operating system PnP code must rout the interrupts. This includes the Legacy Rout bit, Interrupt Rout bit (for each timer), interrupt type (to select the edge or level type for each timer).

The Device Driver code must do the following for an available timer:

1. Set the Overall Enable bit (Offset 04h, bit 0).
2. Set the timer type field (selects one-shot or periodic).
3. Set the interrupt enable.
4. Set the comparator value.

27.3.5 Interrupt Levels

Interrupts directed to the 8259s are active high. See Chapter 19.0, “Interrupts,” for information regarding the polarity programming of the I/O APIC for detecting internal interrupts.

If the interrupts are mapped to the I/O APIC and set for level-triggered mode, they can be shared with PCI interrupts. If more than one timer is configured to share the same IRQ (using the TIMERn_INT_ROUT_CNF fields), then the software must configure the timers to level-triggered mode. Edge-triggered interrupts cannot be shared.

27.3.6 Handling Interrupts

If each timer has a unique interrupt and the timer has been configured for edge-triggered mode, then there are no specific steps required. No read is required to process the interrupt.

If a timer has been configured to level-triggered mode, then its interrupt must be cleared by the software. This is done by reading the interrupt status register and writing a one back to the bit position for the interrupt to be cleared.

Independent of the mode, software can read the value in the main counter to see how time has passed between when the interrupt was generated and when it was first serviced.

If a timer 0 is set up to generate a periodic interrupt, the software can check to see how much time remains until the next interrupt by checking the timer value register.

27.3.7 Issues Related to 64-bit Timers with 32-bit Processors

A 32-bit timer can be read directly using processors that are capable of 32-bit or 64-bit instructions. However, a 32-bit processor may not be able to directly read a 64-bit timer. A race condition occurs if a 32-bit processor reads the 64-bit register using two separate 32-bit reads. The danger is that just after reading one half, the other half rolls over and changes the first half.

If a 32-bit processor needs to access a 64-bit timer, it must first halt the timer before reading both the upper and lower 32 bits of the timer. If a 32-bit processor does not want to halt the timer, it can use the 64-bit timer as a 32-bit timer by setting the TIMERn_32MODE_CNF bit. This will cause the timer to behave as a 32-bit timer. The upper 32 bits will always be zero.

27.3.8 Unloading Device Driver Issues

When unloading device drivers for the HPET (High Precision Event Timer), some precautions may be needed. For example, if the legacy routing is used, when the HPET is disabled, a spurious interrupt could occur. The OS must mask interrupts prior to clearing the LEG_RT_CNF bit.

28.0 Device 29, Function 7: USB 2.0 Host Controller

This section focuses on Intel® 3100 Chipset-specific implementation details of the Universal Serial Bus (USB) Revision 2.0 and Enhanced Host Controller Interface (EHCI) Specification for Universal Serial Bus specifications.

Register address locations not shown in Section 28.2 must be treated as Reserved.

Note: All configuration registers in this section are in the core well and reset by a core well reset and the D3-to-D0 warm reset, except as noted.

28.1 Overview

The Intel® 3100 Chipset contains an Enhanced Host Controller Interface (EHCI) compliant host controller which supports up to four *USB Rev. 2.0 Specification*-compliant root ports. USB 2.0 allows data transfers up to 480 Mbits/s using the same pins as the four USB1 ports. The Intel® 3100 Chipset contains port-routing logic that determines whether a USB port is controlled by one of the UHCI controllers or by the EHCI controller. A USB 2.0-based Debug Port is also implemented in the Intel® 3100 Chipset. A summary of the key architectural differences between the USB 1.1 UHCI host controllers and the USB 2.0 EHCI host controller is shown in the table below:

Table 816. USB 1.1 and USB 2.0 Comparison

	USB 1.1 UHCI	USB 2.0 EHCI
Accessible by	I/O space	Memory Space
Memory Data Structure	Single linked list	Separated in to Periodic and Asynchronous lists
Differential Signaling Voltage	3.3 V	400 mV
Controllers	2	1
Ports per Controller	2	4

28.2 USB 2.0 Configuration Registers

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 817. USB 2.0 Configuration Registers Summary Table

Offset		Symbol	Register Name/Function	Default	Special Notes	Access
Start	End					
USB 2.0 Configuration Registers						
00h	01h	VID	Vendor ID Register	8086h		RO
02h	03h	DID	Device ID Register	268Ch		RO
04h	05h	CMD	Command Register	0000h		RW
06h	07h	DSR	Device Status Register	0290h		RW
08h	08h	RID	Revision ID Register	See Desc		RO
09h	09h	PI	Programming Interface Register	20h		RO
0Ah	0Ah	SCC	Sub Class Code Register	03h		RO
0Bh	0Bh	BCC	Base Class Code Register	0Ch		RO
0Dh	0Dh	MLT	Master Latency Timer Register	00h		RO
10h	13h	MBAR	Memory Base Address Register	00000000h		RW
2Ch	2Dh	SSVID	Subsystem Vendor ID Register	XXXXh	no h/w reset	RWS
2Eh	2Fh	SSID	Subsystem ID Register	XXXXh	no h/w reset	RWS
34h	34h	CAP_PTR	Capabilities Pointer Register	50h		RO
3Ch	3Ch	ILINE	Interrupt Line Register	00h		RW
3Dh	3Dh	IPIN	Interrupt Pin Register	see register description		RO
50h	50h	PM_CID	Power Management Capability ID Register	01h		RO
51h	51h	PM_NEXT	Next Item Ptr #1 Register	58h		RWS
52h	53h	PM_CAP	Power Mgt Capabilities Register	C9C2h		RO, RWS
54h	55h	PM_CS	Power Mgt Control/Status Register	0000h	2 bits in Suspend	RW
58h	58h	DP_CID	Debug Port Capability ID Register	0Ah		RO
59h	59h	DP_NEXT	Next Item Ptr #2	00h		RO
5Ah	5Bh	DP_BASE	Debug Port Base Offset Register	20A0h		RO
60h	60h	SBRN	USB Release Number Register	20h		RO
61h	61h	FLA	Frame Length Adjustment Register	20h	Suspend	RW
62h	63h	PWC	Port Wake Capabilities Register	01FFh	Suspend	RW
68h	6Bh	ULSEC	USB 2.0 Legacy Support Extended Capability Register	00000001h	Suspend	RW
6Ch	6Fh	ULSCS	USB 2.0 Legacy Support Control/ Status Register	00000000h	Suspend	RW
70h	73h	ISU2SMI	Intel Specific USB 2.0 SMI Register	00000000h	Suspend	RW
80h	80h	AC	Access Control Register	00h		RW, RO
F8h	FBh	MANID	Manufacturer's ID Register	00010F80h		

Notes:

1. "Read/Write Special" means that the register is normally read-only but may be written when the WRT_RDONLY bit is set. Because these registers are expected to be programmed by BIOS during initialization, their contents must not get modified by HCRESET or D3-to-D0 internal reset.
2. All configuration registers in this section are in the core well unless otherwise noted. All configuration registers in this section are reset by the core well reset, and the D3-to-D0 warm reset, unless otherwise noted.
3. "Suspend" means that the register is implemented in the Suspend power well.
4. All bits or registers not listed are reserved. All reserved bits read as 0, but must be ignored by software.
5. All bits reset to 0 unless otherwise indicated.

28.2.1 Register Details

28.2.1.1 Offset 00 - 01h: VID – Vendor ID Register

Table 818. Offset 00 - 01h: VID – Vendor ID Register

Device: 29	***Function:*** 7
Offset: 00 - 01h	***Size:*** 16 bit
Default Value: 8086h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:00	VID	**Vendor ID:** This is a 16-bit value assigned to Intel.	8086h	RO

28.2.1.2 Offset 02 - 03h: DID – Device Identification Register

Table 819. Offset 02 - 03h: DID – Device Identification Register

Device: 29	***Function:*** 7
Offset: 02 - 03h	***Size:*** 16 bit
Default Value: 268C	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:00	DID	**Device ID:** This is a 16-bit value assigned to the USB2 host controller.	268C	RO

28.2.1.3 Offset 04 - 05h: CMD – Command Register

Table 820. Offset 04 - 05h: CMD – Command Register (Sheet 1 of 2)

Device: 29	***Function:*** 7
Offset: 04 - 05h	***Size:*** 16 bit
Default Value: 0000h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:11	Reserved	Reserved	00000b	
10	INT_DIS	**Interrupt Disable:** 0 = The function is capable of generating interrupts. 1 = The function can not generate its interrupt to the interrupt controller. The corresponding Interrupt Status bit is not affected by the interrupt enable. This bit defaults to '0'. This bit is added as part of the *Conventional PCI 2.3 Specification*.	0	R/W
09	FBE	**Fast Back to Back Enable:** Reserved as '0'.	0	

Table 820. Offset 04 - 05h: CMD – Command Register (Sheet 2 of 2)

Device: 29	**Function:** 7
Offset: 04 - 05h	**Size:** 16 bit
Default Value: 0000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
08	SERR#_EN	**SERR# Enable:** 0 = The EHC is disabled from generating (internally) SERR# 1 = The EHC is capable of generating (internally) SERR# in the following cases: • Reception of status other than "Successful" on a memory read completion (if SERR on Aborts Enable is also set) • Detection of an address or command parity error and the Parity Error Response bit is set • Detection of a data parity error (when the data is going to the EHC) and the Parity Error Response bit is set • Since USB 2.0 logic does not support parity checking, bit 6 is never set.	0	R/W
07	WCC	**Wait Cycle Control:** Reserved as '0'.	0	
06	PER	**Parity Error Response:** Reserved as '0'.	0	
05	VPS	**VGA Palette Snoop:** Reserved as '0'.	0	
04	PMWE	**Postable Memory Write Enable:** Reserved as '0'.	0	
03	SCE	**Special Cycle Enable:** Reserved as '0'.	0	
02	BME	**Bus Master Enable:** 0 = Clearing the BME bit shuts down the EHC DMA engines in the same manner that clearing the Run/Stop does. However, the schedule status bits and the HCHalted bit do not change based on the BME value 1 = Acts as a master on the PCI bus for USB transfers. Notes on the EHC implementation: • Writes to change this bit occur immediately. Specifically, a write followed by a read will return the updated value. • When the BME bit is changed from 1 to 0, the EHC will cease accessing main memory within 2 microframes (250 µs). During this time, any number of reads and/or writes to memory may occur.	0	R/W
01	MSE	**Memory Space Enable:** This bit controls access to the USB 2.0 Memory Space registers. 0 = Accesses to the USB 2.0 registers are disabled 1 = Accesses to the USB 2.0 registers are enabled. The Base Address register for USB 2.0 must be programmed before this bit is set.	0	R/W
00	IOSE	**I/O Space Enable:** Reserved as '0'.	0	

28.2.1.4 Offset 06 - 07h: DSR – Device Status Register

Note: For the writable bits, software must write a one to clear bits that are set. Writing a zero to the bit has no effect.

Table 821. Offset 06 - 07h: DSR – Device Status Register (Sheet 1 of 2)

Device: 29	***Function:*** 7
Offset: 06 - 07h	***Size:*** 16 bit
Default Value: 0290h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15	DPE	**Detected Parity Error:** 0 = No SERR# detected. 1 = Set when a parity error is detected on the internal interface to the USB host controller, regardless of the setting of bit 6 or bit 8 in the Command register or any other conditions. Software clears this bit by writing a '1' to this bit location.	0	RWC
14	SSE	**Signaled System Error:** 0 = No SERR# detected. 1 = This bit is set whenever it signals SERR# (internally). The SERR_EN bit (bit 8 in the Command Register) must be 1 for this bit to be set. The following conditions can cause the generation of SERR#: • A parity error is seen on address, command, or data (if the data was targeting the EHC) on the internal interface to the USBe host controller due to a parity error on the system interface and bit 6 of the Command register is set to 1. • An EHC-initiated memory read results in a completion packet with a status other than successful on the system interface (if SERR on Aborts Enable is also set to 1). Software clears this bit by writing a '1' to this bit location.	0	RWC
13	RMA	**Received Master-Abort Status:** 0 = No master abort received by EHC on a memory access. 1 = This bit is set when USB 2.0, as a master, receives a master-abort status on a memory access. This is treated as a Host Error and halts the DMA engines. This event can optionally generate an SERR# by setting the SERR# Enable bit and the SERR on Aborts Enable (bit 3, offset 84h). Software clears this bit by writing a '1' to this bit location.	0	RWC
12	RTA	**Received Target Abort Status:** 0 = No target abort received by EHC on memory access. 1 = This bit is set when USB 2.0, as a master, receives a target abort status on a memory access. This is treated as a Host Error and halts the DMA engines. This event can optionally generate an SERR# by setting the SERR# Enable bit and the SERR on Aborts Enable (bit 3, offset 84h). Software clears this bit by writing a '1' to this bit location.	0	RWC
11	STA	**Signaled Target-Abort Status:** This bit is used to indicate when the USB 2.0 function responds to a cycle with a target abort. This should never occur, so this bit is hard-wired to '0'.	0	RO
10:09	DEVT	**DEVSEL# Timing Status:** This 2-bit field defines the timing for DEVSEL# assertion.	01	RO
08	DPD	**Master Data Parity Error Detected:** 0 = No data parity error detected on USB 2.0 read completion packet. 1 = This bit is set whenever a data parity error is detected on a USBe read completion packet on the internal interface to the USBe host controller and bit 6 of the Command register is set to 1. Software clears this bit by writing a '1' to this bit location.	0	RWC
07	FB2BC	**Fast Back-to-Back Capable:** Reserved as '1'.	1	
06	Reserved	**User Definable Features:** Reserved as '0'.	0	

Table 821. Offset 06 - 07h: DSR – Device Status Register (Sheet 2 of 2)

Device: 29	**Function:** 7
Offset: 06 - 07h	**Size:** 16 bit
Default Value: 0290h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
05	C66	**66 MHz Capable:** Reserved as '0'.	0	
04	CLIST	**Capabilities List:** Hardwired to '1' indicating that offset 34h contains a valid capabilities pointer.	1	RO
03	IS	**Interrupt Status:** This read-only bit reflects the state of this function's interrupt at the input of the enable/disable logic. 0 = This bit will be 0 when the interrupt is deasserted. 1 = This bit is a 1 when the interrupt is asserted. The value reported in this bit is independent of the value in the Interrupt Enable bit. This bit is added as part of the *PCI 2.3 Specification*.	0	RO
02:00	Reserved	Reserved	000	

28.2.1.5 Offset 08h: RID – Revision ID Register

Table 822. Offset 08h: RID – Revision ID Register

Device: 29	**Function:** 7
Offset: 08h	**Size:** 8 bit
Default Value: See Desc	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
07:00	RID	**Revision ID:** The value reported in this register depends on the value written to the Revision ID in Device 31, Function 0.	See Desc	RO

28.2.1.6 Offset 09h: PI – Programming Interface Register

Table 823. Offset 09h: PI – Programming Interface Register

Device: 29	**Function:** 7
Offset: 09h	**Size:** 8 bit
Default Value: 20h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
07:00	PIC	**Program Interface Conforms**: A value of 20h indicates that this USB 2.0 Host Controller conforms to the *EHCI Specification*.	20h	RO

28.2.1.7 Offset 0Ah: SCC – Sub Class Code Register

Table 824. Offset 0Ah: SCC – Sub Class Code Register

Device: 29	***Function:*** 7
Offset: 0Ah	***Size:*** 8 bit
Default Value: 03h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	SCC	**Sub Class Code**: A value of 03h indicates that this is a Universal Serial Bus Host Controller.	03h	RO

28.2.1.8 Offset 0Bh: BCC – Base Class Code Register

Table 825. Offset 0Bh: BCC – Base Class Code Register

Device: 29	***Function:*** 7
Offset: 0Bh	***Size:*** 8 bit
Default Value: 0Ch	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	BCC	**Base Class Code:** A value of 0Ch indicates that this is a Serial Bus controller.	0Ch	RO

28.2.1.9 Offset 0Dh: MLT – Master Latency Timer Register

Table 826. Offset 0Dh: MLT – Master Latency Timer Register

Device: 29	***Function:*** 7
Offset: 0Dh	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	MLT	**Master Latency Timer:** Because the USB 2.0 controller is internally implemented with arbitration on an internal interface, it does not need a master latency timer. The bits will be fixed at 0.	00h	RO

28.2.1.10 Offset 10 - 13h: MBAR – Memory Base Address Register

Table 827. Offset 10 - 13h: MBAR – Memory Base Address Register

Device: 29	***Function:*** 7
Offset: 10 - 13h	***Size:*** 32 bit
Default Value: 00000000h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:10	BA	**Base Address:** Bits [31:10] correspond to memory address signals [31:10], respectively. This gives 1 KByte of relocatable memory space aligned to 1 KByte boundaries.	00000h	R/W
09:04	Reserved	**Reserved**	00h	
03	PREF	**Prefetchable:** This bit is hardwired to 0 indicating that this range must not be prefetched.	0	RO
02:01	TPE	**Type:** This field is hardwired to 0 indicating that this range can be mapped anywhere within 32-bit address space.	00	RO
00	RTE	**Resource Type Indicator:** This bit is hardwired to 0 indicating that the base address field in this register maps to memory space.	0	RO

28.2.1.11 Offset 2C - 2Dh: SSVID – USB 2.0 Subsystem Vendor ID Register

Table 828. Offset 2C - 2Dh: SSVID – USB 2.0 Subsystem Vendor ID Register

Device: 29	***Function:*** 7
Offset: 2C - 2Dh	***Size:*** 16 bit
Default Value: XXXXh (See description)	***Power Well:*** Core
Reset: *None*	

Bits	Name	Description	Reset Value	Access
15:00	SSVID	**Subsystem Vendor ID:** This register, in combination with the USB 2.0 Subsystem ID register, enables the operating system to distinguish each subsystem from the others. Writes to this register are enabled when the WRT_RDONLY bit (offset 80h, bit 0) is set to 1.	XXXXh	R/W

28.2.1.12 Offset 2E - 2Fh: SSID – USB 2.0 Subsystem ID Register

Table 829. Offset 2E - 2Fh: SSID – USB 2.0 Subsystem ID Register

Device: 29	***Function:*** 7
Offset: 2E - 2Fh	***Size:*** 16 bit
Default Value: XXXXh (See description)	***Power Well:*** Core
Reset: *None*	

Bits	Name	Description	Reset Value	Access
15:00	SSID	**Subsystem ID:** BIOS sets the value in this register to identify the Subsystem ID. This register, in combination with the Subsystem Vendor ID register, enables the operating system to distinguish each subsystem from other(s). Writes to this register are enabled when the WRT_RDONLY bit (offset 80h, bit 0) is set to 1. Writes must be done as a single 16-bit cycle.	XXXXh	R/W

28.2.1.13 Offset 34h: CAP_PTR – Capabilities Pointer Register

Table 830. Offset 34h: CAP_PTR – Capabilities Pointer Register

Device: 29	***Function:*** 7
Offset: 34h	***Size:*** 8 bit
Default Value: 50h	***Power Well:*** Core
Lockable:	

Bits	Name	Description	Reset Value	Access
07:00	CAP_PTR	**Capabilities Pointer:** This register points to the starting offset of the USB 2.0 capabilities ranges.	50	RO

28.2.1.14 Offset 3Ch: ILINE – Interrupt Line Register

Table 831. Offset 3Ch: ILINE – Interrupt Line Register

Device: 29	***Function:*** 7
Offset: 3Ch	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** *Core*
Lockable: D3-to-D0	

Bits	Name	Description	Reset Value	Access
07:00	ILINE	**Interrupt line:** This data is not used. It is used as a scratchpad register to communicate to software the interrupt line that the interrupt pin is connected to.	00h	RW

28.2.1.15 Offset 3Dh: IPIN – Interrupt Pin Register

Table 832. Offset 3Dh: IPIN – Interrupt Pin Register

Device: 29	***Function:*** 7
Offset: 3Dh	***Size:*** 8 bit
Default Value: See Description	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	IPIN	**Interrupt pin:** Bits 03:00 reflect the value of D29IP.EIP in configuration space. Bits 07:04 are hardwired to 0000b.	See Description	RO

28.2.1.16 Offset 50h: PM_CID – PCI Power Management Capability ID Register

Table 833. Offset 50h: PM_CID – PCI Power Management Capability ID Register

Device: 29	***Function:*** 7
Offset: 50h	***Size:*** 8 bit
Default Value: 01h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	PM_CID	**PCI Power Management Capability ID:** A value of 01h indicates that this is a PCI Power Management capabilities field.	01h	RO

28.2.1.17 Offset 51h: PM_NEXT – Next Item Pointer #1 Register

Table 834. Offset 51h: PM_NEXT – Next Item Pointer #1 Register

Device: 29	***Function:*** 7
Offset: 51h	***Size:*** 8 bit
Default Value: 58h	***Power Well:*** *Core*
Lockable: *Not D3-to-DO*	

Bits	Name	Description	Reset Value	Access
07:00	PM_NEXT	**Next Item Pointer:** This register defaults to 58h, which indicates that the next capability registers begin at configuration offset 58h. This register is writable when the WRT_RDONLY bit is set. This allows BIOS to effectively hide the Debug Port capability registers, if necessary. This register must only be written during system initialization before the plug-and-play software has enabled any master-initiated traffic. Only values of 58h (Debug Port visible) and 00h (Debug Port invisible) are expected to be programmed in this register.	58h	R/W

28.2.1.18 Offset 52 - 53h: PM_CAP – Power Management Capabilities Register

Normally, this register is read-only to report capabilities to the power management software. In order to report different power management capabilities depending on the system in which the Intel® 3100 Chipset is used, bits 15:11 and 08:06 in this register are writable when the WRT_RDONLY bit is set. The value written to this register does not affect the hardware other than changing the value returned during a read.

Reset: core well, but not D3-to-D0.

Table 835. Offset 52 - 53h: PM_CAP – Power Management Capabilities Register

Device: 29	***Function:*** 7
Offset: 52 - 53h	***Size:*** 16 bit
Default Value: C9C2h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:11	PME_Support	This 5-bit field indicates the power states in which the function may assert PME#. The Intel® 3100 Chipset EHC does not support the D1 or D2 states. For all other states, the Intel® 3100 Chipset EHC is capable of generating PME#. Software must never need to modify this field.	11001b	R/W-Special
10	D2_Support	The D2 state is not supported. It is hardwired to '0'.	0b	RO
09	D1_Support	The D1 state is not supported.It is hardwired to '0'.	0b	RO
08:06	Aux_Current	EHC reports 375 mA maximum Suspend well current required when in the $D3_{COLD}$ state. This value can be written by BIOS when a more accurate value is known.	111b	R/W-Special
05	DSI	Reports 0, indicating that no device-specific initialization is required. It is hardwired to '0'.	0b	RO
04	Reserved	Reserved	0b	
03	PME Clock	Reports 0, indicating that no PCI clock is required to generate PME#. It is hardwired to '0'.	0b	RO
02:00	Version	Reports 010, indicating that it complies with Revision 1.1 of the *PCI Power Management Specification*. It is hardwired to '010'.	010	RO

28.2.1.19 Offset 54 - 55h: PM_CS – Power Management Control/Status Register

Table 836. Offset 54 - 55h: PM_CS – Power Management Control/Status Register (Sheet 1 of 2)

Device: 29	***Function:*** 7
Offset: 54 - 55h	***Size:*** 16 bit
Default Value: 0000h	***Reset (bits 15:08):*** Suspend Well, not D3-to-D0 nor core well reset
	Reset (bits 1:0): *Core Well*

Bits	Name	Description	Reset Value	Access
15	PME_Status	0 = Writing a 0 has no effect. 1 = Set when EHC would normally assert the PME# signal independent of the state of the PME_En bit. ***Note:*** Writing a 1 to this bit will clear it and cause the internal PME to deassert (if enabled). This bit must be explicitly cleared by the operating system each time the operating system is loaded.	0	RWC
14:13	Data_Scale	Hardwired to "00" because it does not support the associated Data register.	00	RO
12:09	Data_Select	Hardwired to "0000" because it does not support the associated Data register.	0000	RO

Table 836. Offset 54 - 55h: PM_CS – Power Management Control/Status Register (Sheet 2 of 2)

Device: 29	***Function:*** 7
Offset: 54 - 55h	***Size:*** 16 bit
Default Value: 0000h	***Reset (bits 15:08):*** Suspend Well, not D3-to-D0 nor core well reset
	Reset (bits 1:0): *Core Well*

Bits	Name	Description	Reset Value	Access
08	PME_En	A '1' enables EHC to generate an internal PME signal when PME_Status is '1'. This bit must be explicitly cleared by the operating system each time it is initially loaded.	0	RW
07:02	Reserved	Reserved	00h	
01:00	PowerState	This 2-bit field is used both to determine the current power state of EHC function and to set a new power state. The definition of the field values are: 00b – D0 state 11b – $D3_{HOT}$ state If software attempts to write a value of 10b or 01b in to this field, the write operation must complete normally; however, the data is discarded and no state change occurs. When in the D3hot state, accesses to the EHC memory range must not be accessible, but the configuration space must still be accessible. When not in the D0 state, the generation of the interrupt output is blocked. Specifically, the PIRQ[H] is not asserted by the Intel® 3100 Chipset when not in the D0 state. When software changes this value from the $D3_{HOT}$ state to the D0 state, an internal warm (soft) reset is generated, and software must reinitialize the function.	00	RW

28.2.1.20 Offset 58h: DP_CID – Debug Port Capability ID Register

Table 837. Offset 58h: DP_CID – Debug Port Capability ID Register

Device: 29	***Function:*** 7
Offset: 58h	***Size:*** 8 bit
Default Value: 0Ah	***Power Well:*** Core
Reset: *NA*	

Bits	Name	Description	Reset Value	Access
07:00	DP_CID	**Debug Port Capability ID:** This register is hardwired to 0Ah which indicates that this is the start of a Debug Port Capability structure.	0Ah	RO

28.2.1.21 Offset 59h: DP_NEXT – Next Item Pointer #2 Register

Table 838. Offset 59h: DP_NEXT – Next Item Pointer #2 Register

Device: 29	***Function:*** 7
Offset: 59h	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Core
Reset: *NA*	

Bits	Name	Description	Reset Value	Access
07:00	DP_NEXT	This register is hardwired to 00h, which indicates there are no more capability structures in this function.	00h	RO

28.2.1.22 Offset 5A - 5Bh: DP_BASE – Debug Port Base Offset Register

This register is hardwired to 20A0h, which indicates that the Debug Port Registers begin at offset A0h in the USB 2.0 function's memory space.

Table 839. Offset 5A - 5Bh: DP_BASE – Debug Port Base Offset Register

Device: 29	***Function:*** 7
Offset: 5A - 5Bh	***Size:*** 16 bit
Default Value: 20A0h	***Power Well:*** Core
Reset: *NA*	

Bits	Name	Description	Reset Value	Access
15:13	BNBR	**BAR Number:** This field is hardwired to 001b to indicate the memory BAR at offset 10h in the EHCI configuration space.	001	RO
12:00	DPO	**Debug Port Offset:** This field is hardwired to 0A0h to indicate that the debug port registers begin at offset A0h in the EHCI memory range.	0A0h	RO

28.2.1.23 Offset 60h: SBRN – Serial Bus Release Number Register

Table 840. Offset 60h: SBRN – Serial Bus Release Number Register

Device: 29	***Function:*** 7
Offset: 60h	***Size:*** 8 bit
Default Value: 20h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	SBRN	**Serial Bus Release Number:** A value of 20h indicates that this controller follows USB 2.0.	20h	RO

28.2.1.24 Offset 61h: FLA – Frame Length Adjustment Register

This feature is used to adjust any offset from the clock source that generates the clock that drives the SOF counter. When a new value is written into these six bits, the length of the frame is adjusted. Its initial programmed value is system dependent based on the accuracy of hardware USB clock and is initialized by system BIOS. This register must only be modified when the *HChalted* bit in the USBSTS register is a one. Changing value of this register while the host controller is operating yields undefined results. It

must not be reprogrammed by USB system software unless the default or BIOS programmed values are incorrect, or the system is restoring the register while returning from a suspended state.

Table 841. Offset 61h: FLA – Frame Length Adjustment Register

Device: 29	**Function:** 7
Offset: 61h	**Size:** 8 bit
Default Value: 20h	**Power Well:** *Suspend*
Reset: Suspend well, not D3-to-D0 nor core well reset	

Bits	Name	Description	Reset Value	Access
07:06	Reserved	Reserved.	00	
05:00	FLTV	**Frame Length Timing Value:** Each decimal value change to this register corresponds to 16 high-speed bit times. The SOF cycle time (number of SOF counter clock periods to generate a SOF microframe length) is equal to 59488 + value in this field. The default value is decimal 32 (20h), which gives a SOF cycle time of 60000. **Frame Length** (# High Speed bit times) (decimal) — FLADJ Value (decimal) 59488 — 0 (00h) 59504 — 1 (01h) 59520 — 2 (02h) … 59984 — 31 (1Fh) 60000 — 32 (20h) … 60480 — 62(3Eh) 60496 — 63 (3Fh)	100000	RW

28.2.1.25 Offset 62 - 63h: PWC – Port Wake Capability Register

This register is in the suspend power well. The intended use of this register is to establish a policy about which ports are to be used for wake events. Bit positions 1-8 in the mask correspond to a physical port implemented on the current EHCI controller. A one in a bit position indicates that a device connected below the port can be enabled as a wake-up device and the port may be enabled for disconnect/connect or overcurrent events as wake-up events. This is an information-only mask register. The bits in this register DO NOT affect the actual operation of the EHCI host controller. The system-specific policy can be established by BIOS initializing this register to a system-specific value. System software uses the information in this register when enabling devices and ports for remote wake-up.

Table 842. Offset 62 - 63h: PWC – Port Wake Capability Register (Sheet 1 of 2)

Device: 29	**Function:** 7
Offset: 62 - 63h	**Size:** *16 bits*
Default Value: 01FFh	**Power Well:** *Suspend*
Reset: *suspend well, and not D3-to-D0 warm reset nor core well*	

Bits	Name	Description	Reset Value	Access
15:09	Reserved	Reserved.	00h	

Table 842. Offset 62 - 63h: PWC – Port Wake Capability Register (Sheet 2 of 2)

Device: 29	**Function:** 7
Offset: 62 - 63h	**Size:** 16 bits
Default Value: 01FFh	**Power Well:** Suspend
Reset: suspend well, and not D3-to-D0 warm reset nor core well	

Bits	Name	Description	Reset Value	Access
08:05	Reserved	Reserved	Fh	
04:01	PWU	**Port Wake Up Capability Mask:** Bit positions 1 through 4 correspond to a physical port implemented on this host controller. For example, bit position 1 corresponds to port 0, position 2 port 1, etc.	Fh	RW
00	PWI	**Port Wake Implemented:** 0 = Indicates that this register is not supported by software. 1 = Indicates that this register is supported by software.	1	RW

28.2.1.26 Offset 64 - 65h: CUO – Classic USB Override Register

This 16-bit register provides a bit corresponding to each of the ports on the EHCI host controller (The *EHCI Specification* supports up to 16 ports). When a bit is set to ‘1’, the corresponding USB port is routed to the classic (UHCI) host controller and will only operate using the classic signaling rates. The feature is implemented with the following requirements:

- The associated Port Owner bit does *not* reflect the value in this Override register. This guarantees compatibility with EHCI drivers.
- BIOS must only write to this register during initialization (while the Configured Flag is ‘0’).
- The register is implemented in the Suspend well to maintain port routing when the core power goes down.
- When a ‘1’ is present in the Override register, then the classic controller operates the port regardless of the EHCI port routing logic. The corresponding EHCI port will always appear disconnected in this mode.

Note: EHCI test modes will not work on a port that has been overridden by this register.

- Port 0 must never be programmed to the Classic USB Override mode. This is because the Debug Port is used on Port 0 and the two modes conflict with each other.

28.2.1.27 Offset 68 - 6Bh: ULSEC – USB 2.0 Legacy Support Extended Capability Register

Table 843. Offset 68 - 6Bh: ULSEC – USB 2.0 Legacy Support Extended Capability Register

Device: 29	***Function:*** 7
Offset: 68 - 6Bh	***Size:*** 32 bit
Default Value: 00000001h	***Power Well:*** *Suspend*
Reset: *suspend well, and not D3-to-D0 warm reset nor core well*	

Bits	Name	Description	Reset Value	Access
31:25	Reserved	Reserved. Hardwired to 00h.	00h	
24	HC_OS	**Host Controller OS Owned Semaphore:** System software sets this bit to request ownership of the EHCI controller. 0 = Ownership of the EHCI controller is not obtained. 1 = Ownership is obtained when this bit reads as 1 and the HC BIOS Owned Semaphore bit reads as clear.	0	RW
23:17	Reserved	Reserved	00h	
16	HC_BIOS	**Host Controller BIOS Owned Semaphore:** The BIOS sets this bit to establish ownership of the EHCI controller. System BIOS will clear this bit in response to a request for ownership of the EHCI controller by system software.	0	RW
15:08	NEHCI	**Next EHCI Capability Pointer:** A value of 00h indicates that there are no EHCI Extended Capability structures in this device.	00h	RO
07:00	CAPID	**Capability ID:** A value of 01h indicates that this EHCI Extended Capability is the Legacy Support Capability.	01h	RO

28.2.1.28 Offset 6C - 6Fh: ULSCS – USB 2.0 Legacy Support Control/Status Register

Writing a '1' to that bit location clears bits that are marked as Read/Write-Clear (RWC).

Table 844. Offset 6C - 6Fh: ULSCS – USB 2.0 Legacy Support Control/Status Register (Sheet 1 of 2)

Device: 29	**Function:** 7
Offset: 6C - 6Fh	**Size:** 32 bit
Default Value: 00000000h	**Power Well:** *Suspend*
Lockable: *Suspend well, and not D3-to-D0 warm reset nor core well*	

Bits	Name	Description	Reset Value	Access
31	SMI_BAR	**SMI on BAR:** 0 = Base Address Register (BAR) not written. 1 = This bit is set to 1 when the Base Address Register (BAR) is written.	0	RWC
30	SMI_PCMD	**SMI on PCI Command:** This bit is set to '1' whenever the PCI Command Register is written. 0 = PCI Command (PCICMD) Register Not written. 1 = This bit is set to 1 when the PCI Command (PCICMD) Register is written.	0	RWC
29	SMI_OSC	**SMI on OS Ownership Change:** 0 = No HC OS Owned Semaphore bit change. 1 = This bit is set to 1 when the HC OS Owned Semaphore bit in the LEG_EXT_CAP register (D29:F7:68h, bit 24) transitions from 1 to 0 or 0 to 1.	0	RWC
28:22	Reserved	Reserved. Hardwired to 0.	00h	
21	SMI_AA	**SMI on Async Advance:** Shadow bit of the *Interrupt on Async Advance* bit in the USB 2.0STS register. To clear this bit, system software must write a one to the *Interrupt on Async Advance* bit in the USB 2.0STS register.	0	RO
20	SMI_HSE	**SMI on Host System Error:** Shadow bit of *Host System Error* bit in the USB 2.0STS. To clear this bit, system software must write a one to the *Host System Error* bit in the USB 2.0STS register.	0	RO
19	SMI_FLR	**SMI on Frame List Rollover:** Shadow bit of *Frame List Rollover* bit in the USB 2.0STS register. To clear this bit, system software must write a one to the *Frame List Rollover* bit in the USB 2.0STS register.	0	RO
18	SMI_PCD	**SMI on Port Change Detect:** Shadow bit of *Port Change Detect* bit in the USB 2.0STS register. To clear this bit, system software must write a one to the *Port Change Detect* bit in the USB 2.0STS register.	0	RO
17	SMI_USBER	**SMI on USB Error:** Shadow bit of *USB Error Interrupt* (USBERRINT) bit in the USB 2.0STS register. To clear this bit, system software must write a one to the *USB Error Interrupt* bit in the USB 2.0STS register.	0	RO
16	SMI_USBC	**SMI on USB Complete:** Shadow bit of *USB Interrupt* (USBINT) bit in the USB 2.0STS register. To clear this bit, system software must write a one to the *USB Interrupt* bit in the USB 2.0STS register.	0	RO
15	SMI_BAREN	**SMI on BAR Enable:** 0 = Disable. 1 = Enable. When this bit is 1 and SMI on BAR (D29:F7:6Ch, bit 31) is 1, then the host controller will issue an SMI.	0	RW
14	SMI_PCIEN	**SMI on PCI Command Enable:** 0 = Disable. 1 = Enable. When this bit is 1 and SMI on PCI Command (D29:F7:6Ch, bit 30) is 1, then the host controller will issue an SMI.	0	RW

Table 844. Offset 6C - 6Fh: ULSCS – USB 2.0 Legacy Support Control/Status Register (Sheet 2 of 2)

Device: 29	**Function:** 7
Offset: 6C - 6Fh	**Size:** 32 bit
Default Value: 00000000h	**Power Well:** *Suspend*
Lockable: *Suspend well, and not D3-to-D0 warm reset nor core well*	

Bits	Name	Description	Reset Value	Access
13	SMI_OSEN	**SMI on OS Ownership Enable:** 0 = Disable. 1 = Enable. When this bit is a 1 AND the OS Ownership Change bit (D29:F7:6Ch, bit 29) is 1, the host controller will issue an SMI.	0	RW
12:06	Reserved	Reserved. Hardwired to 0.	00h	
05	SMI_AAEN	**SMI on Async Advance Enable:** 0 = Disable. 1 = Enable. When this bit is a 1, and the SMI on Async Advance bit (D29:F7:6Ch, bit 21) is a 1, the host controller will issue an SMI immediately.	0	RW
04	SMI_HSEN	**SMI on Host System Error Enable:** 0 = Disable. 1 = Enable. When this bit is a 1, and the SMI on Host System Error (D29:F7:6Ch, bit 20) is a 1, the host controller will issue an SMI.	0	RW
03	SMI_FLREN	**SMI on Frame List Rollover Enable:** 0 = Disable. 1 = Enable. When this bit is a 1, and the SMI on Frame List Rollover bit (D29:F7:6Ch, bit 19) is a 1, the host controller will issue an SMI.	0	RW
02	SMI_PCEN	**SMI on Port Change Enable:** 0 = Disable. 1 = Enable. When this bit is a 1, and the SMI on Port Change Detect bit (D29:F7:6Ch, bit 18) is a 1, the host controller will issue an SMI.	0	RW
01	SMI_USBEN	**SMI on USB Error Enable:** 0 = Disable. 1 = Enable. When this bit is a 1, and the SMI on USB Error bit (D29:F7:6Ch, bit 17) is a 1, the host controller will issue an SMI immediately.	0	RW
00	SMI_USBCE	**SMI on USB Complete Enable:** 0 = Disable. 1 = Enable. When this bit is a 1, and the SMI on USB Complete bit (D29:F7:6Ch, bit 16) is a 1, the host controller will issue an SMI immediately.	0	RW

28.2.1.29 Offset 70 - 73h: ISU2SMI – Intel Specific USB 2.0 SMI Register

This register provides a mechanism for BIOS to provide USB 2.0 related bug fixes and workarounds. Writing a ‘1’ to that bit location clears bits that are marked as Read/Write/Clear (RW/C). Software must clear all SMI status bits prior to setting the global SMI enable bit and individual SMI enable bit to prevent spurious SMI when returning from a power down.

Table 845. Offset 70 - 73h: ISU2SMI – Intel Specific USB 2.0 SMI Register (Sheet 1 of 2)

Device: 29	**Function:** 7
Offset: 70 - 73h	**Size:** 32 bit
Default Value: 00000000h	**Power Well:** Suspend
Lockable: Suspend well, and not D3-to-D0 warm reset nor core well	

Bits	Name	Description	Reset Value	Access
31:30	Reserved	Reserved. Hardwired to 0.	00	
29:22	SMI_PO	**SMI on PortOwner:** Bits 29:22 correspond to the Port Owner bits for ports 1 (22) through 8 (29). These bits are set to '1' whenever the associated Port Owner bits transition from 0->1 or 1->0. Software clears these bits by writing a one.	00000000	RWC
21	SMI_ PMCSR	**SMI on PMCSR:** 0 = Power State bits not modified. 1 = Software modified the Power State bits in the Power Management Control/Status (PMCSR) register (D29:F7:54h).	0	RWC
20	SMI_ASYNC	**SMI on Async:** 0 = No Async Schedule Enable bit change 1 = Async Schedule Enable bit transitioned from 1 to 0 or 0 to 1.	0	RWC
19	SMI_PER	**SMI on Periodic:** 0 = No Periodic Schedule Enable bit change. 1 = Periodic Schedule Enable bit transitions from 1 to 0 or 0 to 1.	0	RWC
18	SMI_CF	**SMI on CF:** 0 = No Configure Flag (CF) change. 1 = Configure Flag (CF) transitions from 1 to 0 or 0 to 1.	0	RWC
17	SMI_HCH	**SMI on HCHalted:** 0 = HCHalted did not transition to 1 (as a result of the Run/Stop bit being cleared). 1 = HCHalted transitions to 1 (as a result of the Run/Stop bit not being cleared).	0	RWC
16	SMI_HCR	**SMI on HCReset:** 0 = HCRESET did not transition to 1. 1 = HCRESET transitioned to 1.	0	RWC
15:14	Reserved	Reserved.	0	
13:06	SMI_POEN	**SMI on PortOwner Enable:** When any of these bits are '1' and the corresponding SMI on PortOwner bits are '1', then the host controller will issue an SMI. Unused ports must have their corresponding bits cleared.	00000000	RW
05	SMI_PMSCRE N	**SMI on PMSCR Enable:** 0 = Disable. 1 = Enable. When this bit is 1 and SMI on PMSCR is 1, then the host controller will issue an SMI.	0	RW
04	SMI_ASYEN	**SMI on Async Enable:** 0 = Disable. 1 = Enable. When this bit is 1 and SMI on Async is 1, then the host controller will issue an SMI.	0	RW
03	SMI_PEREN	**SMI on Periodic Enable:** 0 = Disable. 1 = Enable. When this bit is 1 and SMI on Periodic is 1, then the host controller will issue an SMI.	0	RW

Table 845. Offset 70 - 73h: ISU2SMI – Intel Specific USB 2.0 SMI Register (Sheet 2 of 2)

Device: 29	**Function:** 7
Offset: 70 - 73h	**Size:** *32 bit*
Default Value: *00000000h*	**Power Well:** *Suspend*
Lockable: Suspend well, and not D3-to-D0 warm reset nor core well	

Bits	Name	Description	Reset Value	Access
02	SMI_CFEN	**SMI on CF Enable:** 0 = Disable. 1 = Enable. When this bit is 1 and SMI on CF is 1, then the host controller will issue an SMI.	0	RW
01	SMI_HCHEN	**SMI on HCHalted Enable:** 0 = Disable. 1 = Enable. When this bit is a 1 and SMI on HCHalted is 1, then the host controller will issue an SMI.	0	RW
00	SMI_HCREN	**SMI on HCReset Enable:** 0 = Disable. 1 = Enable. When this bit is a 1 and SMI on HCReset is 1, then host controller will issue an SMI.	0	RW

28.2.1.30 Offset 80h: AC – Access Control Register

Table 846. Offset 80h: AC – Access Control Register

Device: 29	**Function:** 7
Offset: 80h	**Size:** *8 bit*
Default Value: 00h	**Power Well:**

Bits	Name	Description	Reset Value	Access
07:01	Reserved	Reserved	00h	
00	WRT_RDONLY	**Write Read only:** 0 = Disables a select group of normally read-only registers in the EHC function to be written by software. 1 = Enables a select group of normally read-only registers in the EHC function to be written by software. Registers that may only be written when this mode is entered are noted in the summary tables and detailed description as “Read/Write-Special”. The registers fall into two categories: a. System-configured parameters b. Status bits	0	RW

28.2.1.31 Offset F8 - FBh: MANID – Manufacturer's ID Register

Table 847. Offset F8 - FBh: MANID – Manufacturer's ID Register

Device: 29	***Function:*** 7
Offset: F8 - FBh	***Size:*** 32 bit
Default Value: 00010F80h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:24	Reserved	Reserved	00h	
23:16	SID	**Stepping ID:** This field increments for each stepping of the part. This field can be used by software to differentiate steppings when the Revision ID may not change. Implementation Note: A single Stepping ID can be implemented that is readable from all functions in the chip because all of them are incremented in lock-step. See Device 31, Function 0, Offset F8h for the reported value.	01h - A1	RO
15:08	MID	**Manufacturing ID:** Indicates 0Fh = Intel	0Fh	RO
07:00	Reserved	Reserved	80h	

28.3 Memory-Mapped I/O Registers

The USB 2.0 EHCI memory-mapped I/O space is composed of two sets of registers: Capability Registers and Operational Registers.

The EHCI controller does not support as a target memory transactions that are locked transactions. Attempting to access the EHCI controller Memory-Mapped I/O space using locked memory transactions will result in undefined behavior.

Note: When the USB 2.0 function is in the D3 PCI power state, accesses to the USB 2.0 memory range are ignored and will result in a master abort. Similarly, if the Memory Space Enable (MSE) bit is not set in the Command register in configuration space, the memory range will not be decoded by the Enhanced Host Controller (EHC). If the MSE bit is not set, then the Intel® 3100 Chipset must default to allowing any memory accesses for the range specified in the BAR to go to PCI. This is because the range may not be valid and, therefore, the cycle must be made available to any other targets that may be currently using that range.

28.3.1 Host Controller Capability Registers

These registers specify the limits, restrictions and capabilities of the host controller implementation.

Within the Host Controller Capability Registers, only the Structural Parameters register is writable. This register is implemented in the Suspend well and is only reset by the standard suspend-well hardware reset, not by HCRESET or the D3-to-D0 reset.

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 848. Host Controller Capability Registers Summary Table

MEM_BASE+ Offset		Symbol	Register Name/Function	Default	Special Notes	Access
Start	End					
00h	00h	CAPLENGTH	Capabilities Length Register	20h		RO
02h	03h	HCIVERSION	Host Controller Interface Version Number Register	0100h		RO
04h	07h	HCSPARAMS	Structural Parameters Register	00104208h	Suspend	RWS
08h	0Bh	HCCPARAMS	Capability Parameters Register	00006871h		RO

Notes:
1. "Suspend" means that the register is implemented in the Suspend power well
2. "Read/Write Special" means that the register is normally read-only, but may be written when the WRT_RDONLY bit is set. Because these registers are expected to be programmed by BIOS during initialization, their contents must not get modified by HCRESET or D3-to-D0 internal reset.

28.3.1.1 Offset 00h: CAPLENGTH – Capability Length Register

This register is used as an offset to add to the Memory Base Register to find the beginning of the Operational Register Space. This is fixed at 20h, indicating that the Operation Registers begin at offset 20h.

Table 849. Offset 00h: CAPLENGTH – Capability Length Register

Device: 29 ***Function:*** 7
Offset: 00h ***Size:*** 8 bit
Default Value: 20h ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	CRLV	**Capability Register Length Value:** This register is used as an offset to add to the Memory Base Register (D29:F7:10h) to find the beginning of the Operational Register Space. This field is hardwired to 20h indicating that the Operation Registers begin at offset 20h.	20h	RO

28.3.1.2 Offset 02 - 03h: HCIVERSION – Host Controller Interface Version Number Register

This is a 2-byte register containing a BCD encoding of the version number of interface to which this host controller interface conforms.

Table 850. Offset 02 - 03h: HCIVERSION – Host Controller Interface Version Number Register

Device: 29 ***Function:*** 7
Offset: 02 - 03h ***Size:*** 16 bit
Default Value: 0100h ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:00	HCIVERSION	**Host Controller Interface Version Number:** This is a two-byte register containing a BCD encoding of the version number of interface to which this host controller interface conforms.	0100h	RO

28.3.1.3 Offset 04 - 07h: HCSPARAMS – Host Controller Structural Parameters Register

This is a set of fields that are structural parameters: Number of downstream ports, etc. Some fields in this register are writable when the WRT_RDONLY bit is set. Fields that are described as "hardwired" are never writable. This register is implemented in the suspend well to avoid having to reload the parameters following a system sleep state in which the core power is removed.

Table 851. Offset 04 - 07h: HCSPARAMS – Host Controller Structural Parameters Register

Device: 29	***Function:*** 7
Offset: 04 - 07h	***Size:*** 32 bit
Default Value: 00104208h	***Power Well:*** *Suspend*
Reset: Suspend well reset, but not D3-to-D0 reset or HCRESET	

Bits	Name	Description	Reset Value	Access
31:24	Reserved	Reserved.	0	
23:20	DP_N	**Debug Port Number:** Hardwired to 1h, indicating that the Debug Port is on the lowest numbered port.	1	RO
19:17	Reserved	Reserved.	0	
16	P_INDICATOR	**Port Indicators:** This bit indicates whether the ports support port indicator control. The Intel® 3100 Chipset USB 2.0 Controller does not support Port Indicator LEDs, and this bit is hard wired to '0'.	0	RWS
15:12	N_CC	**Number of Companion Controllers:** This field indicates the number of companion controllers associated with this USB 2.0 host controller. A zero in this field indicates there are no companion host controllers. Port-ownership hand-off is not supported. Only high-speed devices are supported on the host controller root ports. A value larger than one in this field indicates there are companion USB 1.1 host controller(s). Port-ownership hand-offs are supported. High, Full- and Low-speed devices are supported on the host controller root ports. The Intel® 3100 Chipset allows the default value of 4h to be overwritten by BIOS. When removing classic controllers, they must be disabled in the following order: Function 1, and Function 0, which correspond to ports 4:3, and 2:1, respectively.	0100	RWS
11:08	N_PCC	**Number of Ports per Companion Controller:** This field indicates the number of ports supported per companion host controller. It is used to indicate the port routing configuration to system software. Hardwired to 2h.	0010	Read Only
07:04	Reserved	Reserved	00	
03:00	N_PORTS	This field specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register Space. Valid values are in the range of 1H to FH. 4h is reported by default. However, software may write a value less than 8 for some platform configurations. A zero in this field is undefined.	0100	RWS

28.3.1.4 Offset 08 - 0Bh: HCCPARAMS – Host Controller Capability Parameters Register

This register provides general mode information that affects the generation of the data structure in memory.

Table 852. Offset 08 - 0Bh: HCCPARAMS – Host Controller Capability Parameters Register

Device: 29	***Function:*** 7
Offset: 08 - 0Bh	***Size:*** 32 bit
Default Value: 00006871h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0000h	
15:08	EECP	**EHCI Extended Capabilities Pointer:** This field is hardwired to 68h, indicating that the EHCI capabilities list exists and begins at offset 68h in the PCI configuration space.	68h	RO
07:03	Reserved	Reserved	01110	
02	ASPC	**Asynchronous Schedule Park Capability:** This bit is hardwired to 0 indicating that the Host Controller does not support this optional feature.	0	RO
01	PFLF	**Programmable Frame List Flag:** 0 = If this bit is set to a zero, then system software must use a frame list length of 1024 elements with this host controller. The USBCMD register *Frame List Size* field is a read-only register and must be set to zero. 1 = If set to a one, then system software can specify and use a smaller frame list and configure the host controller via the USBCMD register *Frame List Size* field. The frame list must always be aligned on a 4K page boundary. This requirement ensures that the frame list is always physically contiguous. Different frame list lengths are not supported. This bit is read-only '0'.	0	RO
00	ADD_CAP	**64-bit Addressing Capability:** This field documents the addressing range capability of this implementation. The value of this field determines whether software must use the 32-bit or 64-bit data structures. Values for this field have the following interpretation: 0 = Data structures using 32-bit address memory pointers 1 = Data structures using 64-bit address memory pointers Only 64-bit addressing is supported. This bit is read-only '1'. Only 44 bits of addressing is supported. Bits 63:44 will always be 0 on cycles generated to memory.	1	RO

28.3.2 Host Controller Operational Registers

This section defines the enhanced host controller operational registers. These registers are located after the capabilities registers. The operational register base must be DWord aligned and is calculated by adding the value in the first capabilities register to the base address of the enhanced host controller register address space. In the following text, the offset is relative to the Memory Base Register. All registers are 32 bits in length. Software must read and write these registers using only Dword accesses.

These registers are divided into two sets. The first set at offsets 20h to 3Fh are implemented in the core power well. Unless otherwise noted, the core-well registers are reset by the assertion of any of the following:

- core well hardware reset
- HCRESET

- D3-to-D0 reset

The second set at offsets 60h to the end of the implemented register space are implemented in the Suspend power well. Unless otherwise noted, the suspend power well registers are reset by the assertion of either of the following:

- suspend well hardware reset
- HCRESET

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 853. Host Controller Operational Registers Summary Table

MEM_BASE + Offset		Symbol	Register Name/Function	Default	Special Notes	Access
Start	End					
20h	23h	USB 2.0CMD	USB 2.0 Command Register	00080000h		RW
24h	27h	USB 2.0STS	USB 2.0 Status Register	00001000h		RWC, RO
28h	2Bh	USB 2.0INTR	USB 2.0 Interrupt Enable Register	00000000h		RW
2Ch	2Fh	FRINDEX	USB 2.0 Frame Index Register	00000000h		RW, RO
30h	33h	CTRLDSSEGMENT	Control Data Structure Segment Register	00000000h		RW
34h	37h	PERIODICLISTBASE	Period Frame List Base Address Register	00000XXXh		RW
38h	3Bh	ASYNCLISTADDR	Next Asynchronous List Address Register	00000000h		RW, RO
60h	63h	CONFIGFLAG	Configure Flag Register	00000000h	Suspend	RW, RO
64h	67h	PORTSC	Port 1 Status and Control Register	00003000h	Suspend	RW
68h	6Bh	PORTSC	Port 2 Status and Control Register	00003000h	Suspend	RW
6Ch	6Fh	PORTSC	Port 3 Status and Control Register	00003000h	Suspend	RW
70h	73h	PORTSC	Port 4 Status and Control Register	00003000h	Suspend	RW
A0h	B3h		Debug Port Registers (see Section 28.13.2, "Debug Port Registers")			RW

28.3.2.1 Offset 20 - 23h: USB 2.0CMD – USB 2.0 Command Register

The Command Register indicates the command to be executed by the serial bus host controller. Writing to the register causes a command to be executed.

Table 854. Offset 20 - 23h: USB 2.0CMD – USB 2.0 Command Register (Sheet 1 of 2)

Device: 29	***Function:*** 7
Offset: 20 - 23h	***Size:*** 32 bit
Default Value: 00080000h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:24	Reserved	Reserved.	00h	
23:16	ITC	**Interrupt Threshold Control:** Default 08h. This field is used by system software to select the maximum rate at which the host controller will issue interrupts. The only valid values are defined below. If software writes an invalid value to this register, the results are undefined. Value — Maximum Interrupt Interval 00h — Reserved 01h — 1 microframe 02h — 2 microframes 04h — 4 microframes 08h — 8 microframes (default, equates to 1 ms) 10h — 16 microframes (2 ms) 20h — 32 microframes (4 ms) 40h — 64 microframes (8 ms) Refer to Section 4 in the *EHCI Specification* for interrupts affected by this field.	08h	RW
15:12	Reserved	Reserved	0h	
11:08	UAPM	**Unimplemented Asynchronous Park Mode Bits:** This field is hardwired to 0000b because the host controller does not support this optional feature.	0h	RO
07	LHCR	**Light Host Controller Reset:** This optional reset is not supported and is hardwired to 0.	0	RO
06	IAAD	**Interrupt on Async Advance Doorbell:** This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to *ring* the doorbell. When the host controller has evicted all appropriate cached schedule state, it sets the *Interrupt on Async Advance* status bit in the USBSTS register. If the *Interrupt on Async Advance Enable* bit in the USBINTR register is a one then the host controller will assert an interrupt at the next interrupt threshold. See the *EHCI Specification* for operational details. The host controller sets this bit to a zero after it has set the *Interrupt on Async Advance* status bit in the USBSTS register to a one. Software must not write a one to this bit when the asynchronous schedule is disabled. Doing so will yield undefined results.	0	RW
05	ASY_SCEN	**Asynchronous Schedule Enable:** Default 0b. This bit controls whether the host controller skips processing the Asynchronous Schedule. Values mean: 0 = Do not process the Asynchronous Schedule 1 = Use the ASYNCLISTADDR register to access the Asynchronous Schedule.	0	RW
04	P_SCEN	**Periodic Schedule Enable:** Default 0b. This bit controls whether the host controller skips processing the Periodic Schedule. Values mean: 0 = Do not process the Periodic Schedule 1 = Use the PERIODICLISTBASE register to access the Periodic Schedule.	0	RW
03:02	FLS	**Frame List Size:** Hardwired to 00b because it only supports the 1024-element frame list size.	0	RO

Table 854. Offset 20 - 23h: USB 2.0CMD – USB 2.0 Command Register (Sheet 2 of 2)

Device: 29	**Function:** 7
Offset: 20 - 23h	**Size:** 32 bit
Default Value: 00080000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
01	HCRESET	**Host Controller Reset:** This control bit used by software to reset the host controller. The effects of this on Root Hub registers are similar to a Chip Hardware Reset (i.e., RSMRST# assertion and PWROK deassertion). When software writes a one to this bit, the Host Controller resets its internal pipelines, timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. ***Note:*** PCI Configuration registers and Host Controller Capability Registers are not affected by this reset. All operational registers, including port registers and port state machines are set to their initial values. Port ownership reverts to the companion host controller(s), with the side effects described in the *EHCI Specification*. Software must reinitialize the host controller in order to return the host controller to an operational state. This bit is set to zero by the Host Controller when the reset process is complete. Software cannot terminate the reset process early by writing a zero to this register. Software must not set this bit to a one when the HCHalted bit in the USBSTS register is a zero. Attempting to reset an actively running host controller will result in undefined behavior.	0	RW
00	RS	**Run/Stop:** Default 0b. 1=Run. 0=Stop. 1 = The Host Controller proceeds with execution of the schedule. The Host Controller continues execution as long as this bit is set to a 1. 0 = The Host Controller completes the current and any actively pipelined transactions on the USB and then halts. The Host Controller must halt within 16 microframes after software clears the Run bit. The HC Halted bit in the status register indicates when the Host Controller has finished its pending pipelined transactions and has entered the stopped state. Software must not write a 1 to this field unless the host controller is in the Halted state (i.e., HCHalted in the USBSTS register is a one). The following table explains how the different combinations of Run and Halted must be interpreted: Run/Stop Halted Interpretation 0 0 Valid - in the process of halting 0 1 Valid - halted 1 0 Valid - running 1 1 Invalid - the HCHalted bit clears immediately Memory read cycles initiated by the EHC that receive any status other than Successful will result in this bit being cleared (and also affect the Host Error bit).	0	RW

28.3.2.2 Offset 24 - 27h: USB 2.0STS – USB 2.0 Status Register

This register indicates pending interrupts and various states of the Host Controller. The status resulting from a transaction on the serial bus is not indicated in this register. Software sets a bit to 0 in this register by writing a 1 to it. See the Interrupts description in section 4 of the *EHCI Specification* for additional information concerning USB 2.0 interrupt conditions.

Table 855. Offset 24 - 27h: USB 2.0STS – USB 2.0 Status Register (Sheet 1 of 2)

Device: 29	**Function:** 7
Offset: 24 – 27h	**Size:** 32 bit
Default Value: 00001000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0	
15	ASY_SSTAT	**Asynchronous Schedule Status:** This bit reports the current real status of the Asynchronous Schedule. 0 = The status of the Asynchronous Schedule is disabled. 1 = The status of the Asynchronous Schedule is enabled. The Host Controller is not required to *immediately* disable or enable the Asynchronous Schedule when software transitions the *Asynchronous Schedule Enable* bit in the USBCMD register. When this bit and the *Asynchronous Schedule Enable* bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0).	0	RO
14	PER_SSTAT	**Periodic Schedule Status:** This bit reports the current real status of the Periodic Schedule. 0 = The status of the Periodic Schedule is disabled. 1 = The status of the Periodic Schedule is enabled. The Host Controller is not required to *immediately* disable or enable the Periodic Schedule when software transitions the *Periodic Schedule Enable* bit in the USBCMD register. When this bit and the *Periodic Schedule Enable* bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0).	0	RO
13	RECL	**Reclamation:** This is a read-only status bit, which is used to detect an empty asynchronous schedule. The operational model and valid transitions for this bit are described in Section 4 of the *EHCI Specification*.	0	RO
12	HCH	**HCHalted:** 0 = This bit is a zero whenever the Run/Stop bit is a one. 1 = The Host Controller sets this bit to 1 after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host Controller hardware (e.g., internal error).	1	RO
11:06	Reserved	Reserved	0	
05	INT_ASYA	**Interrupt on Async Advance:** System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a one to the *Interrupt on Async Advance Doorbell* bit in the USBCMD register. This status bit indicates the assertion of that interrupt source.	0	RWC
04	HS_ERR	**Host System Error:** 0 = No serious error occurred during a host system access involving the Host Controller module. 1 = The Host Controller sets this bit to 1 when a serious error occurs during a host system access involving the Host Controller module. Memory read cycles initiated by the EHC that receive any status other than Successful will result in this bit being set. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs. A hardware interrupt is generated to the system (if enabled in the Interrupt Enable Register).	0	RWC
03	FLRO	**Frame List Rollover:** 0 = No *Frame List Index* rollover from its maximum value to 0. 1 = The Host Controller sets this bit to a one when the *Frame List Index* rolls over from its maximum value to zero. Since only 1024-entry Frame List Size is supported, the *Frame List Index* rolls over every time FRNUM[13] toggles.	0	RWC

Table 855. Offset 24 - 27h: USB 2.0STS – USB 2.0 Status Register (Sheet 2 of 2)

Device: 29	**Function:** 7
Offset: 24 – 27h	**Size:** 32 bit
Default Value: 00001000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
02	PCD	**Port Change Detect:** The Host Controller sets this bit to a one when any port for which the *Port Owner* bit is set to zero has a change bit transition from a zero to a one or a Force Port Resume bit transition from a zero to a one as a result of a J-K transition detected on a suspended port. This bit will also be set as a result of the *Connect Status Change* being set to a one after system software has relinquished ownership of a connected port by writing a zero to a port's *Port Owner* bit. This bit is allowed to be maintained in the Auxiliary power well. Alternatively, it is also acceptable that, on a D3 to D0 transition of the EHCI HC device, this bit is loaded with the OR of all of the PORTSC change bits (including Force port resume, overcurrent change, enable/disable change and connect status change). Regardless of the implementation, whenever this bit is readable, i.e., in the D0 state, it must provide a valid view of the Port Status registers.	0	RWC
01	USBERRINT	**USB Error Interrupt:** 0 = No error condition. 1 = The Host Controller sets this bit to 1 when completion of a USB transaction results in an error condition, e.g., error counter underflow. If the TD on which the error interrupt occurred also had its IOC bit set, both this bit and Bit 0 are set. See the *EHCI Specification* for a list of the USB errors that will result in this interrupt being asserted.	0	RWC
00	USBINT	**USB Interrupt:** 0 = No completion of a USB transaction whose Transfer Descriptor had its IOC bit set. No short packet is detected. 1 = The Host Controller sets this bit to 1 on the completion of a USB transaction, which results in the retirement of a Transfer Descriptor that had its IOC bit set. The Host Controller also sets this bit to 1 when a short packet is detected (actual number of bytes received was less than the expected number of bytes).	0	RWC

28.3.2.3 Offset 28- 2Bh: USB 2.0INTR – USB 2.0 Interrupt Enable Register

This register enables and disables reporting of the corresponding interrupt to the software. When a bit is set and the corresponding interrupt is active, an interrupt is generated to the host. Interrupt sources that are disabled in this register still appear in the Status Register to allow the software to poll for events.

Each interrupt enable bit description indicates whether it is dependent on the interrupt threshold mechanism (see Section 4 of the *EHCI Specification*).

Table 856. Offset 28- 2Bh: USB 2.0INTR – USB 2.0 Interrupt Enable Register

Device: 29	**Function:** 7
Offset: 28 – 2Bh	**Size:** 32 bit
Default Value: 00000000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:06	Reserved	Reserved	0	
05	INT_AAEN	**Interrupt on Async Advance Enable:** 0 = Disable. 1 = When this bit is a one, and the *Interrupt on Async Advance* bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the *Interrupt on Async Advance* bit.	0	RW
04	HSE_EN	**Host System Error Enable:** 0 = Disable. 1 = When this bit is a one, and the *Host System Error Status* bit in the USBSTS register is a one, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the *Host System Error* bit.	0	RW
03	FLR_EN	**Frame List Rollover Enable:** 0 = Disable. 1 = When this bit is a one, and the *Frame List Rollover* bit in the USBSTS register is a one, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the *Frame List Rollover* bit.	0	RW
02	PCI_EN	**Port Change Interrupt Enable:** 0 = Disable. 1 = When this bit is a one, and the Port Change Detect bit in the USBSTS register is a one, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the *Port Change Detect* bit.	0	RW
01	USBEI_EN	**USB Error Interrupt Enable:** 0 = Disable. 1 = When this bit is a one, and the USBERRINT bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software by clearing the *USBERRINT* bit.	0	RW
00	USBI_EN	**USB Interrupt Enable:** 0 = Disable. 1 = When this bit is a one, and the USBINT bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software by clearing the *USBINT* bit.	0	RW

Note: For all enable register bits, 1= Enabled, 0= Disabled

28.3.2.4 Offset 2C - 2Fh: FRINDEX – Frame Index Register

This register is used by the host controller to index into the periodic frame list. The register updates every 125 microseconds (once each microframe). Bits [12:3] are used to select a particular entry in the Periodic Frame List during periodic schedule execution. The number of bits used for the index is fixed at 10 since only 1024-entry frame lists are supported. This register must be written as a DWord. Word and byte writes produce undefined results. This register cannot be written unless the Host Controller is in the Halted state as indicated by the *HCHalted* bit (USB 2.0STS register). A write to this register while the Run/Stop bit is set to a one (USB 2.0CMD register) produces undefined results. Writes to this register also affect the SOF value. See Section 4 of the *EHCI Specification* for details.

Table 857. Offset 2C - 2Fh: FRINDEX – Frame Index Register

Device: 29	**Function:** 7
Offset: 2C - 2Fh	**Size:** 32 bit
Default Value: 00000000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:14	Reserved	Reserved	0	
13:00	FLCI	**Frame List Current Index/Frame Number:** The value in this register increments at the end of each time frame (e.g., microframe).Bits [12:3] are used for the Frame List current index. This means that each location of the frame list is accessed eight times (frames or microframes) before moving to the next index.	0	RW

The SOF frame number value for the bus SOF token is derived or alternatively managed from this register. Please refer to Section 4 of the *EHCI Specification* for a detailed explanation of the SOF value management requirements on the host controller. The value of FRINDEX must be 125 µs (1 microframe) ahead of the SOF token value. The SOF value may be implemented as an 11-bit shadow register. For this discussion, this shadow register is 11 bits and is named SOFV. SOFV updates every 8 microframes. (1 ms). An example implementation to achieve this behavior is to increment SOFV each time the FRINDEX[2:0] increments from a zero to a one.

Software must use the value of FRINDEX to derive the current microframe number and to provide the *get* microframe number function required for client drivers. Therefore, the value of FRINDEX and the value of SOFV must be kept consistent if chip is reset or software writes to FRINDEX. Writes to FRINDEX must also *write-through* FRINDEX[13:3] to SOFV[10:0]. In order to keep the update as simple as possible, software must never write a FRINDEX value where the three least significant bits are 111b or 000b.

28.3.2.5 Offset 30 - 33h: CTRLDSSEGMENT – Control Data Structure Segment Register

This 32-bit register corresponds to the most significant address bits [63:32] for all EHCI data structures. Since the 64-bit Addressing Capability field is hardwired in HCCPARAMS to one, then this register is used with the link pointers to construct 64-bit addresses to EHCI control data structures. This register is concatenated with the link pointer from either the PERIODICLISTBASE, ASYNCLISTADDR, or any control data structure link field to construct a 64-bit address. This register allows the host software to locate all control data structures within the same 4 GByte memory segment.

Note: The Intel® 3100 Chipsethas 44-bit addressing internally for both control and data structures. Address bits 63:44 [31:12] are hardwired to zero independent of the setting of this register. The lower 12 address bits 43:32 [11:0] are fully read/write capable for software compatibility and specification compliance.

Table 858. Offset 30 - 33h: CTRLDSSEGMENT – Control Data Structure Segment Register

Device: 29	***Function:*** 7
Offset: 30 – 33h	***Size:*** 32 bit
Default Value: 00000000h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:12	HC_UP	**Upper Address[63:44]:** This 20-bit field is hard wired to zero.	0	RO
11:00	HC_LWR	**Upper Address[43:32]:** This 12-bit field corresponds to address bits 43:32 when forming a control data structure address.	0	RW

28.3.2.6 Offset 34 - 37h: PERIODICLISTBASE – Periodic Frame List Base Address Register

This 32-bit register contains the beginning address of the Periodic Frame List in the system memory. Since the host controller operates in 64-bit mode (as indicated by the one in the 64-bit Addressing Capability field in the HCCSPARAMS register) (offset 08h, bit 0), then the most significant 32 bits of every control data structure address comes from the CTRLDSSEGMENT register. System software loads this register prior to starting the schedule execution by the Host Controller. The memory structure referenced by this physical memory pointer is assumed to be 4 Kbyte aligned. The contents of this register are combined with the Frame Index Register (FRINDEX) to enable the Host Controller to step through the Periodic Frame List in sequence.

Table 859. Offset 34 - 37h: PERIODICLISTBASE – Periodic Frame List Base Address Register

Device: 29	***Function:*** 7
Offset: 34 - 37h	***Size:*** 32 bit
Default Value: 00000XXXh	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:12	HC_LOW	**Base Address (Low):** These bits correspond to memory address signals [31:12], respectively.	0	RW
11:00	Reserved	Reserved. Must be written as 0s. During runtime, the value of these bits are undefined.	XXXh	

28.3.2.7 Offset 38 - 3Bh: ASYNCLISTADDR – Current Asynchronous List Address Register

This 32-bit register contains the address of the next asynchronous queue head to be executed. Since the host controller operates in 64-bit mode (as indicated by a one in 64-bit Addressing Capability field in the HCCPARAMS register), then the most significant 32 bits of every control data structure address comes from the CTRLDSSEGMENT register. Bits [4:0] of this register cannot be modified by system software and will always return zeros when read. The memory structure referenced by this physical memory pointer is assumed to be 32-byte aligned.

Table 860. Offset 38 - 3Bh: ASYNCLISTADDR – Current Asynchronous List Address Register

Device: 29	**Function:** 7
Offset: 38 - 3Bh	**Size:** 32 bit
Default Value: 00000000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:05	LPL	**Link Pointer Low:** These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH).	0	RW
04:00	Reserved	Reserved	0	

28.3.2.8 Offset 60 - 63h: CONFIGFLAG – Configure Flag Register

This register is in the suspend power well. It is only reset by hardware when the suspend power is initially applied or in response to a host controller reset. This 32-bit register contains the address of the next asynchronous queue head to be executed. Since the host controller operates in 64-bit mode (as indicated by a 1 in 64-bit Addressing Capability field in the HCCPARAMS register), then the most significant 32 bits of every control data structure address comes from the CTRLDSSEGMENT register (offset 08h). Bits [4:0] of this register cannot be modified by system software and will always return 0's when read. The memory structure referenced by this physical memory pointer is assumed to be 32-byte aligned.

Table 861. Offset 60 - 63h: CONFIGFLAG – Configure Flag Register

Device: 29	**Function:** 7
Offset: 60 - 63h	**Size:** 32 bit
Default Value: 00000000h	**Power Well:** *Suspend*

Bits	Name	Description	Reset Value	Access
31:01	Reserved	Reserved. Reads from this field will always return 0.	0	
00	CF	**Configure Flag:** Host software sets this bit as the last action in its process of configuring the Host Controller. This bit controls the default port-routing control logic. Bit values and side effects are listed below. See section 4 of the *EHCI Specification* for operation details. 0 = Port routing control logic default-routes each port to the classic host controllers. 1 = Port routing control logic default-routes all ports to this host controller.	0	RW

28.3.2.9 Offset 64 - 67h, 68 - 6Bh, 6C - 6Fh, 70 - 73h: PORTSC – Port *N* Status and Control Register

A host controller must implement one or more port registers. Software uses the N_Port information from the Structural Parameters Register to determine how many ports need to be serviced. All ports have the structure defined below. Software must not write to unreported Port Status and Control Registers.

This register is in the suspend power well. It is only reset by hardware when the suspend power is initially applied or in response to a host controller reset. The initial conditions of a port are:

- No device connected
- Port disabled

Note: When a device is attached, the port state transitions to the connected state and system software will process this as with any status change notification. Refer to Section 4 of the *EHCI Specification* for operational requirements for how change events interact with port suspend mode.

Note: If a port is being used as the Debug Port, then the port may report device connected and enabled when the Configured Flag is a zero.

Table 862. Offset 64 - 67h, 68 - 6Bh, 6C - 6Fh, 70 - 73h: PORTSC – Port N Status and Control (Sheet 1 of 5)

Device: 29 **Function:** 7
Offset: Port 1) 64 - 67h, Port 2 68 - 6Bh Port 3) 6C - 6Fh, Port 470 - 73h **Size:** 32 bit
Default Value: 00003000h **Power Well:** *Suspend*

Bits	Name	Description	Reset Value	Access
31:23	Reserved	Reserved.	000h	
22	WKOC_E	**Wake on Overcurrent Enable:** 0 = Disable. (Default). 1 = Writing this bit to a one enables the port to be sensitive to overcurrent conditions as wake-up events. When enabled to do so, the EHC sets the PME Status bit in the Power Management Control/Status Register (offset 54, bit 15) when the overcurrent Active bit (bit 4 of this register) is set.	0	RW
21	WKDSCNNT_E	**Wake on Disconnect Enable:** 0 = Disable. (Default). 1 = Writing this bit to a one enables the port to be sensitive to device disconnects as wake-up events. When enabled to do so, the EHC sets the PME Status bit in the Power Management Control/Status Register (offset 54, bit 15) when the Current Connect Status changes from connected to disconnected (i.e., bit 0 of this register changes from 1 to 0).	0	RW
20	WKCNNT_E	**Wake on Connect Enable:** 0 = Disable. (Default). 1 = Writing this bit to a one enables the port to be sensitive to device connects as wake-up events. When enabled to do so, the EHC sets the PME Status bit in the Power Management Control/Status Register (offset 54, bit 15) when the Current Connect Status changes from disconnected to connected (i.e., bit 0 of this register changes from 0 to 1).	0	RW
19:16	PT_CTRL	**Port Test Control:** Default = 0000b. When this field is zero, the port is NOT operating in a test mode. A non-zero value indicates that it is operating in test mode and the specific test mode is indicated by the specific value. The encoding of the test mode bits are (0110b - 1111b are reserved): Bits Test Mode 0000b Test mode not enabled 0001b Test J_STATE - During this test mode the hardware will force pre-emphasis disabled to the AFE 0010b Test K_STATE - During this test mode the hardware will force pre-emphasis disabled to the AFE 0011b Test SE0_NAK 0100b Test Packet 0101b Test FORCE_ENABLE Refer to *USB Rev. 2.0 Specification*, Chapter 7 and the *EHCI Specification*, Chapter 4 for details on each test mode. The EHC does not support the option to run the port tests while the Run/Stop bit is a one.	0000	RW
15:14	**Reserved**	**Reserved.**	00	

Table 862. Offset 64 - 67h, 68 - 6Bh, 6C - 6Fh, 70 - 73h: PORTSC – Port N Status and Control (Sheet 2 of 5)

Device: 29	**Function:** 7
Offset: Port 1) 64 - 67h, Port 2 68 - 6Bh Port 3) 6C - 6Fh, Port 470 - 73h	**Size:** 32 bit
Default Value: 00003000h	**Power Well:** *Suspend*

Bits	Name	Description	Reset Value	Access
13	PO	**Port Owner:** Default = 1b. This bit unconditionally goes to a 0b when the *Configure Flag* makes a 0b to 1b transition. This bit unconditionally goes to 1b whenever the *Configure Flag* bit is zero. System software uses this field to release ownership of the port to a selected host controller (in the event that the attached device is not a high-speed device). Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that a companion host controller owns and controls the port. See Section 4 of the *EHCI Specification* for operational details.	1	RW
12	PP	**Port Power:** Hard-wired with a value of one. This indicates that the port does have power.	1	RO
11:10	LS	**Line Status:** These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. These bits are used for detection of low-speed USB devices prior to the port reset and enable sequence. This field is valid only when the port enable bit is zero and the current connect status bit is set to a one. The encoding of the bits is as follows: Bits[11:10] — USB State — Interpretation 00b — SE0 — Not Low-speed device, perform EHCI reset 10b — J-state — Not Low-speed device, perform EHCI reset 01b — K-state — Low-speed device, release ownership of port 11b — Undefined — Not Low-speed device, perform EHCI reset	0	RO
09	Reserved	Reserved.	0	

Table 862. Offset 64 - 67h, 68 - 6Bh, 6C - 6Fh, 70 - 73h: PORTSC – Port N Status and Control (Sheet 3 of 5)

Device: 29	**Function:** 7
Offset: Port 1) 64 - 67h, Port 2 68 - 6Bh Port 3) 6C - 6Fh, Port 470 - 73h	**Size:** 32 bit
Default Value: 00003000h	**Power Well:** *Suspend*

Bits	Name	Description	Reset Value	Access
08	PR	**Port Reset:** 0 = Port is not in Reset (default). 1 = Port is in Reset. When software writes a one to this bit (from a zero), the bus reset sequence as defined in the *USB Rev. 2.0 Specification* is started. Software writes a zero to this bit to terminate the bus reset sequence. Software must keep this bit at a one long enough to guarantee the reset sequence, as specified in the *USB Rev. 2.0 Specification*, completes. ***Note:*** When software writes this bit to a one, it must also write a zero to the *Port Enable* bit. ***Note:*** When software writes a zero to this bit, there may be a delay before the bit status changes to a zero. The bit status will not read as a zero until after the reset has completed. If the port is in high-speed mode after reset is complete, the host controller will automatically enable this port (e.g., set the *Port Enable* bit to a one). A host controller must terminate the reset and stabilize the state of the port within 2 ms of software transitioning this bit from a one to a zero. For example, if the port detects that the attached device is high-speed during reset, then the host controller must have the port in the enabled state within 2 ms of software writing this bit to a zero. The HCHalted bit in the USB2STS register must be a zero before software attempts to use this bit. The host controller may hold Port Reset asserted to a one when the HCHALTED bit is a one. The *Run/Stop* bit in the Command Register must be set in order for the *Port Reset* bit to be cleared.	0	RW
07	PS	**Suspend:** 0 = Port not in suspend state (default). 1 = Port in suspend state. Port Enabled Bit and Suspend bit of this register define the port states as follows: Bits [Port Enabled, Suspend]Port State 0X Disable 10 Enable 11 Suspend When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to a 1. In the suspend state, the port is sensitive to resume detection. The bit status does not change until the port is suspended and there may be a delay in suspending a port if there is a transaction currently in progress on the USB. A write of zero to this bit is ignored by the host controller. The host controller will unconditionally set this bit to a zero when: • Software sets the *Force Port Resume* bit to a zero (from a one). • Software sets the *Port Reset* bit to a one (from a zero). If host software sets this bit to a one when the port is not enabled (i.e., Port enabled bit is a zero) the results are undefined.	0	RW

Table 862. Offset 64 - 67h, 68 - 6Bh, 6C - 6Fh, 70 - 73h: PORTSC – Port N Status and Control (Sheet 4 of 5)

Device: 29 **Function:** 7

Offset: Port 1) 64 - 67h, Port 2 68 - 6Bh Port 3) 6C - 6Fh, Port 470 - 73h **Size:** 32 bit

Default Value: 00003000h **Power Well:** *Suspend*

Bits	Name	Description	Reset Value	Access
06	FPR	**Force Port Resume:** 0 = No resume (K-state) detected/driven on port (default). 1 = Resume detected/driven on port. This functionality defined for manipulating this bit depends on the value of the *Suspend* bit. For example, if the port is not suspended (*Suspend* and *Enabled* bits are a one) and software transitions this bit to a one, then the effects on the bus are undefined. Software sets this bit to a 1 to drive resume signaling. The Host Controller sets this bit to a 1 if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to a one because a J-to-K transition is detected, the *Port Change Detect* bit in the USBSTS register is also set to a one. If software sets this bit to a one, the host controller must not set the *Port Change Detect* bit. When the EHCI controller owns the port, the resume sequence follows the defined sequence documented in the *USB Rev. 2.0 Specification*. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains a one. Software must appropriately time the Resume and set this bit to a zero when the appropriate amount of time has elapsed. Writing a zero (from one) causes the port to return to high-speed mode (forcing the bus below the port into a high-speed idle). This bit will remain a one until the port has switched to the high-speed idle. The host controller must complete this transition within 2 milliseconds of software setting this bit to a zero.	0	RW
05	OCC	**Overcurrent Change:** 0 = No change (default). 1 = This bit gets set to a one when there is a change to the Overcurrent Active bit. Software clears this bit by writing a one to this bit position. The functionality of this bit is not dependent upon the port owner.	0	RWC
04	OCA	**Overcurrent Active:** 0 = This port does not have an overcurrent condition (deafult). 1 = This port currently has an overcurrent condition. This bit will automatically transition from a one to a zero when the overcurrent condition is removed. The functionality of this bit is not dependent upon the port owner. The Intel® 3100 Chipset automatically disables the port when the overcurrent active bit is '1'.	0	RO
03	PEDC	**Port Enable / Disable Change:** 0 = No change (default). 1 = Port enabled/disabled status has changed. For the root hub, this bit gets set to a one only when a port is disabled due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the *USB Specification* for the definition of a port error). This bit is not set due to the Disabled-to-Enabled transition, nor due to a disconnect. Software clears this bit by writing a 1 to it.	0	RWC

Table 862. Offset 64 - 67h, 68 - 6Bh, 6C - 6Fh, 70 - 73h: PORTSC – Port N Status and Control (Sheet 5 of 5)

Device: 29	***Function:*** 7
Offset: Port 1) 64 - 67h, Port 2 68 - 6Bh Port 3) 6C - 6Fh, Port 470 - 73h	***Size:*** 32 bit
Default Value: 00003000h	***Power Well:*** *Suspend*

Bits	Name	Description	Reset Value	Access
02	PENDIS	**Port Enabled / Disabled:** 0 = Disable (default). 1 = Enable. As described in the *EHCI Specification*, ports are enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. The host controller will only set this bit to a one when the reset sequence determines that the attached device is a high-speed device.	0	RW
01	CSC	**Connect Status Change:** 0 = No change (default). 1 = Change in Current Connect Status. Indicates a change has occurred in the port's Current Connect Status. The host controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be "setting" an already-set bit (i.e., the bit will remain set). Software sets this bit to 0 by writing a 1 to it.	0	RWC
00	CCS	**Current Connect Status:** 0 = No device is present (default). 1 = Device is present on port. This value reflects the current state of the port and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set.	0	RO

28.4 EHC Initialization

The following describes the expected EHC initialization sequence in chronological order, beginning with a complete power cycle in which the suspend well and core well have been off.

28.4.1 Power On

The suspend well is a "deeper" power plane than the core well, which means that the suspend well is always functional when the core well is functional but the core well may not be functional when the suspend well is. Therefore, the suspend well reset pin (RSMRST#) deasserts before the core well reset pin (PWROK) rises.

1. The suspend well reset deasserts, leaving all registers and logic in the suspend well in the default state. However, it is not possible to read any registers until after the core well reset deasserts.
2. The core well reset deasserts, leaving all registers and logic in the core well in the default state. The EHC configuration space is accessible at this point. The core well reset can (and typically does) occur without the suspend well reset asserting. This means that all of the Configure Flag and Port Status and Control bits (and any other suspend-well logic) may be in any valid state at this time.

28.4.2 Driver Initialization

See Chapter 4 of the *EHCI Specification, Rev. 1.0*.

28.4.3 EHC Resets

In addition to the standard hardware resets, portions of the EHC are reset by the HCRESET bit and the transition from the D3hot device power management state to the D0 state. The effect of each of these resets are:

Table 863. HCRESET Bit Summary

Reset	Does Reset	Does Not Reset	Comments
HCRESET bit set	Memory space registers except Structural Parameters (which is written by BIOS)	Configuration Registers	The HCRESET must only affect registers that the EHCI driver controls. PCI Configuration space and BIOS-programmed parameters must not be reset.
Software writes the Device Power State from D3hot (11b) to D0 (00b)	Core-well registers (except BIOS-programmed registers)	Suspend-well registers; BIOS-programmed Core-well Registers	The D3-to-D0 transition must not cause wake information (suspend well) to be lost. It also must not clear BIOS-programmed registers because BIOS may not be invoked following the D3-to-D0 transition.

If the detailed register descriptions give exceptions to these rules, those exceptions override these rules. This summary is provided to help explain the reasons for the reset policies.

28.5 Data Structures in Main Memory

See Section 3 and Appendix B of the *EHCI Specification, Rev. 1.0* for details.

28.6 USB 2.0 Enhanced Host Controller DMA

The USB 2.0 enhanced host controller implements three sources of USB packets. They are, in order of priority on USB during each microframe:

- the USB 2.0 Debug Port (see Section 28.13),
- the Periodic DMA engine, and
- the Asynchronous DMA engine.

The Intel® 3100 Chipset always performs any currently-pending debug port transaction at the beginning of a microframe, followed by any pending periodic traffic for the current microframe. If there is time left in the microframe, then the EHC performs any pending asynchronous traffic at the end of the microframe (EOF1). The debug port traffic is only presented on one port (Port #0), while the other ports are idle during this time.

The following subsections describe the policies of the periodic and asynchronous DMA engines.

28.6.1 Periodic List Execution

The Periodic DMA engine contains buffering for two control structures (two transactions). By implementing two entries, the EHC is able to pipeline the memory accesses for the next transaction while executing the current transaction on the USB ports. A multiple-packet, high-bandwidth transaction occupies one of these buffer entries, which means that up to six 1 Kbyte data packets may be associated with the two buffered control structures.

In order to simplify the pipelined implementation that is optimized for normal execution, the EHC does not implement immediate retries on High Bandwidth Interrupt transactions that encounter transaction errors (for ins and outs) or a Data Toggle mismatch (for Interrupt In). This is an optional implementation, but not recommended, by the *USB Specification* (Sections 5.9 and 5.9.1) and the *USB Specification*(4.10.3 of Revision 0.95). The EHC will reattempt the transaction when that qTD is encountered again in the periodic schedule. If successful when reattempted, then the EHC will continue with the multiple packets allowed by the high-bandwidth endpoint during that same microframe.

28.6.1.1 Read Policies for Periodic DMA

The Periodic DMA engine performs memory reads for the following structures:

Table 864. Periodic DMA Engine Memory Reads

Memory Structure	Size (DWORDS)	Comments
Periodic Frame List entry	1	The EHC reads the entry for each microframe. The frame list is not internally cached across microframes.
Frame Span Traversal Node	2	
iTD	23	Only the 64-bit addressing format is supported.
siTD	9	Only the 64-bit addressing format is supported.
qTD	13	Only the 64-bit addressing format is supported.
Queue Head	17	Only the 64-bit addressing format is supported.
Out Data	Up to 257	Large read requests are broken down into smaller aligned read requests based on the setting of the Read Request Maximum Length field.

Periodic DMA read policies:

1. The EHC Periodic DMA Engine (PDE) does not generate accesses to main memory unless all three of the following conditions are met:
   a. The HCHalted bit is 0 (memory space, offset 24h, bit 12). Software clears this bit indirectly by setting the RUN/STOP bit to 1.
   b. The Periodic Schedule Status bit is 1 (memory space, offset 24h, bit 14). Software sets this bit indirectly by setting the Periodic Schedule Enable Bit to 1.
   c. The Bus Master Enable bit is 1 (configuration space, offset 04h, bit 2).
2. Once the above conditions are met, the PDE waits until the frame index counter rolls over from the end of microframe 6 to the beginning of microframe 7 to begin prefetching for microframe 0 of the next frame. This means the initial memory access may be delayed up to 1 ms after the DMA-enabled conditions are met. Further delays within the arbitration and datapath are also possible before the first read request is presented on the IMCH/IICH link.
3. The Periodic Frame List Entry is always read from memory before any data structures associated with the new microframe are accessed.

4. Prefetching is limited to the current and next microframes only. If prefetching is disabled, the periodic DMA engine will perform transactions serially (no pipelining) and will read structures for the current microframe only.
5. The PDE fetches structures in the periodic list until all information (including data) is available to run one USB transaction before beginning to fetch the structures for a pipelined transaction. For High-Bandwidth Out transactions, all of the data may not fit into the Data FIFO; in those cases, the next pipelined control structure fetches will be delayed until some data is delivered to USB.
6. The PDE does not refetch the control structure between “Multi” packets of a High Bandwidth endpoint.
7. The PDE will not generate any control structure reads (including the frame list index) if both of the transaction buffers are occupied. Data reads are the only read requests that will be generated by the PDE in this case.
8. The PDE will not pipeline fetch a control structure (iTD, siTD, or QH) if the other transaction slot contains that control structure already. This is to avoid executing based on stale fields in the control structure since a status write (or overlay) is expected to occur following execution of the pending transaction. The PDE will traverse the schedule (periodic frame list entry and any inactive control structures for the microframe) before encountering the Link Pointer to the stale structure. At that point the fetching pauses until the pending transaction is completed. *The iTD could be refetched since a separate status is maintained for each microframe; the PDE will not attempt this optimization*.
9. Once the PDE checks the length of a periodic packet against the remaining time in the microframe (late-start check) and decides that there is not enough time to run it on the wire, then the EHC switches over to run asynchronous traffic. The EHC does not attempt to look for any shorter packets in the remainder of the periodic schedule that might be able to fit in the current microframe.
10. The PDE implements a “Gross Late-Start” check which determines whether any more control or data structure reads will be initiated for transactions associated with the current microframe. The threshold for this check is determined by the Gross Late Start Cut-Off field in configuration register offset 84h.
11. An entry in the 2-deep command FIFO becomes available for a new transaction fetch when any of the following events occur:
    a. The final transaction results are posted in write buffers to memory.
    b. Either of the late-start checks fail for this transaction (or the preceding transaction in the same microframe).
    c. A High-Bandwidth Interrupt transaction times out. For High-Bandwidth Interrupt transactions that time out, the Intel® 3100 Chipset does not immediately retry the transaction as recommended by the USB Specification (Section 5.9.1). Instead, all control and data structures are flushed and the transaction is reattempted the next time that endpoint is scheduled.

Note: When a host error occurs, the commands are kept in the PDE. The EHCI software driver must assert the HCRESET in order to clear the pending transactions before reenabling the PDE.

12. Data fetches are not initiated unless there is room in the Out Data FIFO to consume the amount of data requested.
13. Read requests are broken up and throttled based on the Read Request Maximum Length field and the Request Rate Throttle fields in the configuration register at offset FCh. Control or Data structures that cross a Maximum Length-aligned boundary in memory are broken into multiple requests. This allows other packets from within the IICH to be interleaved on the IMCH/IICH link and through the memory controller to avoid temporary starvation of those functions. When generating the multiple read requests, the EHC will naturally-align the requests

(i.e., 64-byte requests will not fetch across 64-byte address boundaries in memory). This guarantees that, as cache-line sizes increase, the back-to-back requests do not cause double-snoops on specific cache lines. Unlike control structure read requests, only reads for data will be subject to the Request Rate Throttle.

14. Asynchronous DMA memory accesses may be interleaved at any point with the periodic DMA memory accesses on the IMCH/IICH link.

28.6.1.2 Write Policies for Periodic DMA

The Periodic DMA engine performs writes to the following data structures:

Periodic DMA write policies:

1. The Periodic DMA Engine (PDE) will only generate writes after a transaction is executed on USB. Some important notes associated with this rule are:
   a. If either of the late-start checks fails before the transaction is run on the USB ports, then none of the writes normally associated with that transaction will occur. High-Bandwidth Exception: If the late-start check fails after the first packet of a High-Bandwidth (multi) transaction is executed but before the last packet, then the PDE must write the status for any completed transfers to memory.
   b. The Queue Head Overlay write occurs after the first transaction for a qTD is completed on the USB interface.
2. Status writes are always performed after In Data writes for the same transaction.
3. When writing the status back to the two siTDs associated with a backpointer, the PDE first writes to the siTD which was referenced by the backpointer and secondly writes to the siTD which contains the backpointer.
4. Asynchronous DMA memory accesses may be interleaved at any point with the periodic DMA memory accesses on the IMCH/IICH link.
5. When writing back the qTD information after clearing the *Active* bit, the *EHCI Specification* does not require that the *C_Page* field is written. However, due to byte-granular write control, the EHC does write to this field, and the value is not necessarily the final or incremented *C_Page* value.

28.6.2 Asynchronous List Execution

The Asynchronous DMA engine contains buffering for two control structures (two transactions). By implementing two entries, the EHC is able to pipeline the memory accesses for the next transaction while executing the current transaction on the USB ports.

28.6.2.1 Read Policies for Asynchronous DMA

The Asynchronous DMA engine performs reads for the following structures:

Table 865. Asynchronous DMA Engine Reads

Memory Structure	Size (DWORDS)	Comments
qTD	13	Only the 64-bit addressing format is supported.
Queue Head	17	Only the 64-bit addressing format is supported.
Out Data	Up to 129	Large read requests are broken down in to smaller aligned read requests based on the setting of the Read Request Maximum Length field.

Asynchronous DMA read policies:

1. The EHC Asynchronous DMA Engine (ADE) does not generate accesses to main memory unless all four of the following conditions are met. (The ADE may be active when the periodic schedule is actively executed, unlike the description in the *EHCI Specification*; since the EHC contains independent DMA engines, the ADE may perform memory accesses interleaved with the PDE accesses.)
   a. The HCHalted bit is 0 (memory space, offset 24h, bit 12). Software clears this bit indirectly by setting the RUN/STOP bit to 1.
   b. The Asynchronous Schedule Status bit is 1 (memory space, offset 24h, bit 14). Software sets this bit indirectly by setting the Asynchronous Schedule Enable Bit to 1.
   c. The Bus Master Enable bit is 1 (configuration space, offset 04h, bit 2).
   d. The ADE is not sleeping due to the detection of an empty schedule. There is not one single bit that indicates this state. However, the sleeping state is entered when the Queue Head with the H bit set is encountered when the Reclamation bit in the USB 2.0 Status register is 0.
2. Once the above conditions are met, the ADE immediately begins reading the Queue Head to which the Current Asynchronous List Address Register points. Delays within the arbitration and datapath are possible before the first read request is presented on the IMCH/IICH link.
3. If prefetching is disabled, the ADE will perform transactions serially (no pipelining).
4. The ADE fetches structures in the asynchronous list until all information (including data) is available to run one USB transaction before beginning to fetch the structures for a pipelined transaction.
5. The ADE will not generate any control structure reads if both of the transaction buffers are occupied. Data reads are the only read requests that will be generated by the ADE in this case.
6. The ADE does not fetch data when a QH is encountered in the Ping state. An Ack handshake in response to the Ping results in the ADE writing the QH to the Out state, which results in the fetching and delivery of the Out Data on the next iteration through the asynchronous list.
7. The ADE will not pipeline fetch a Queue Head if the other transaction slot contains that Queue Head already (i.e., only one active QH). This is to avoid executing based on stale fields in the Queue Head since a status write (or overlay) is expected to occur following execution of the pending transaction. The ADE will traverse the schedule (any inactive Queue Heads) before encountering the Link Pointer to the stale structure. At that point the fetching pauses until the pending transaction is completed.
8. Once the ADE checks the length of an asynchronous packet against the remaining time in the microframe (late-start check) and decides that there is not enough time to run it on the wire, then the EHC stops all activity on the USB ports for the remainder of that microframe. The EHC does not attempt to look for any shorter packets in the remainder of the asynchronous schedule that might be able to fit in the current microframe. Unlike the PDE, the ADE keeps the transaction internally for executing in the next microframe without refetching from memory.
9. An entry in the 2-deep command FIFO becomes available for a new transaction fetch when any of the following events occur:
   a. The final transaction results are posted in write buffers to memory.
   b. A host error causes an unexpected halt. Any unexecuted transactions in the command FIFO are flushed.

10. Once the ADE detects an “empty” asynchronous schedule as described in Section 4 of the *EHCI Specification*, it implements a waking mechanism like the one in the example. The amount of time that the ADE “sleeps” is 10 µs +/-30 ns.
11. Data fetches are not initiated unless there is room in the Out Data FIFO for the amount of data requested.
12. Read requests are broken up and throttled based on the Read Request Maximum Length field and the Request Rate Throttle fields in the configuration register at offset FCh. Control or Data structures that cross a Maximum Length-aligned boundary in memory are broken into multiple requests. This allows other packets from within the IICH to be interleaved on the IMCH/IICH link and through the memory controller to avoid temporary starvation of those functions. When generating the multiple read requests, the EHC will naturally-align the requests (i.e., 64-byte requests will not fetch across 64-byte address boundaries in memory). This guarantees that, as cache-line sizes increase, the back-to-back requests do not cause double-snoops on specific cache lines. Unlike control structure read requests, only reads for data will be subject to the Request Rate Throttle.
13. Periodic DMA memory accesses may be interleaved at any point with the asynchronous DMA memory accesses on IMCH/IICH link.

28.6.2.2 Write Policies for Asynchronous DMA

The Asynchronous DMA engine performs writes to the following memory structures:

Table 866. Asynchronous DMA Engine Writes

Memory Structure	Size (DWORDS)	Comments
Asynchronous Queue Head Overlay	14	Only the 64-bit addressing format is supported. Dwords 0Ch through 43h are written.
Asynchronous Queue Head Status Write	3	Dwords 14h through 1Fh are written.
Asynchronous qTD Status Write	3	Dwords 04h through 0Fh are written. PID Code, IOC, Buffer Pointer (page 0), and Alt. Next qTD Pointer are rewritten with the original value.
In Data	Up to 129	Data writes are broken down into 16 Dword-aligned chunks.

Asynchronous DMA write policies:

1. The Asynchronous DMA Engine (ADE) will only generate writes after a transaction is executed on USB. Some important notes associated with this rule are:
   a. If the late-start check fails before the transaction is run on the USB ports, then the USB transaction and the writes are delayed until the next opportunity to run the asynchronous traffic.
   b. The Queue Head Overlay write occurs after the first transaction for a qTD is completed on the USB interface.
2. Status writes are always performed after In Data writes for the same transaction.
3. Periodic DMA memory accesses may be interleaved at any point with the Asynchronous DMA memory accesses on IMCH/IICH link.
4. When writing back the qTD information after clearing the *Active* bit, the *EHCI Specification* does not require that the *C_Page* field is written. However, due to byte-granular write control, the EHC does write to this field, and the value is not necessarily the final or incremented *C_Page* value.

28.7 Data Encoding and Bit Stuffing

See Chapter 8 of the *USB Rev. 2.0 Specification.*

28.8 Packet Formats

See Chapter 8 of the *USB Rev. 2.0 Specification.*

28.9 USB 2.0 Interrupts and Error Conditions

Section 4 of the *EHCI Specification* goes into detail on the EHC interrupts and the error conditions that cause them. All error conditions that the EHC detects can be reported through the EHCI Interrupt status bits. Only Intel® 3100 Chipset-specific interrupt and error-reporting behavior is documented in this section. The EHCI Interrupts Section (in Section 4 of the *EHCI Specification*) must be read first, followed by this section, to fully comprehend the EHC interrupt and error-reporting functionality.

- Based on the EHC's Buffer sizes and buffer management policies, the Data Buffer Error can never occur.
- Master Abort and Target Abort responses from the system interface on EHC-initiated read packets will be treated as Fatal Host Errors. The EHC halts when these conditions are encountered.
- The Intel® 3100 Chipset may assert the interrupts that are based on the interrupt threshold as soon as the status for the last complete transaction in the interrupt interval has been posted in the internal write buffers. The requirement in the *EHCI Specification* (that the status is written to memory) is met internally, even though the write may not be seen on the IMCH/IICH interface before the interrupt is asserted.
- Since the Intel® 3100 Chipset only supports the 1024-element Frame List size, the Frame List Rollover interrupt occurs every 1024 milliseconds.
- The Intel® 3100 Chipset delivers interrupts using PIRQ#[H].
- The Intel® 3100 Chipset does not modify the CERR count on an Interrupt IN when the "Do Complete-Split" execution criteria are not met.
- For complete-split transactions in the Periodic list, the "Missed Microframe" bit does not get set on a control-structure-fetch that fails the late-start test. If subsequent accesses to that control structure do not fail the late-start test, then the "Missed Microframe" bit will get set and written back.

28.9.1 Aborts on USB 2.0-Initiated Memory Reads

If a read initiated by the EHC receives any status other than "Successful" in the completion packet, the EHC treats it as a fatal host error. The following actions are taken when this occurs:

- The Host System Error status bit is set
- The DMA engines are halted, the Run/Stop bit is cleared, and the HCHalted bit is set, after completing up to one more transaction on the USB interface
- If enabled (by the Host System Error Enable), then an interrupt is generated
- If the status is Master Abort, then the Received Master Abort bit in configuration space is set
- If the status is Target Abort, then the Received Target Abort bit in configuration space is set

- If enabled (by the SERR Enable bit and the SERR on Abort Enable bit in the function's configuration space), then the Signaled System Error bit in configuration bit is set and the internal SERR signal is asserted

28.9.2 Host Interface Parity Errors

In the event of parity errors on the host-side interface, the EHC is required to respond as shown in the following table.

Table 867. Host Interface Parity Errors (Sheet 1 of 2)

Input Scenario			Resulting Behavior			
Event	Parity Error Resp	SERR# En (CMD register, bit 8)	DPE (DSR register, bit 15)	Master DPE (DSR register, bit 8)	Host System Error (USB Status)	Notes
Downbound Request Command Parity Error	0	X	1	0	0	Do take the cycle, as normal.
	1	0	1	0	1	Do not take the cycle (master abort). No SERR# generated
	1	1	1	0	1	Do not take the cycle (master abort). SERR# generated
Downbound Request Address Parity Error	Identical to the Command Parity Error rows above					
Downbound Request Data Parity Error	0	X	1	0	0	Do take the cycle, as normal.
	1	0	1	0	1	Cycle is taken. Halt the Host Controller, if currently not halted. Drop the write data. No SERR# generated. (No PERR# on the IMCH/IICH link) Software can only determine that the error occurred through the Host Error interrupt or by polling.
	1	1	1	0	1	Cycle is taken. Halt the Host Controller, if currently not halted. Drop the write data. SERR# generated. (No PERR# on the IMCH/IICH link)
Downbound Completion Command Parity Error	This must be treated the same as the Downbound Request Command Parity Error because the error could be on the completion/request bit.					

Table 867. Host Interface Parity Errors (Sheet 2 of 2)

Input Scenario			Resulting Behavior			
Event	Parity Error Resp	SERR# En (CMD register, bit 8)	DPE (DSR register, bit 15)	Master DPE (DSR register, bit 8)	Host System Error (USB Status)	Notes
Downbound Completion Data Parity Error	0	X	1	0	0	Do take the cycle, as normal.
	1	0	1	1	1	Cycle is taken. Halt the Host Controller, if currently not halted. Do not forward data to the USB ports. No SERR# generated. (No PERR# on the IMCH/IICH link) Software can only determine that the error occurred through the Host Error interrupt or by polling.
	1	1	1	1	1	Cycle is taken. Halt the Host Controller, if currently not halted. Do not forward data to the USB ports. SERR# generated. (No PERR# on the IMCH/IICH link)

The EHC is accessible as a target after the parity errors are detected (assuming that the system instability has not caused a deadlock in some way).

All of the above description is required behavior. The following text describes the Intel® 3100 Chipset-specific implementation details:

There are three general forms of parity errors that the EHC may detect on its system interface:

- Command/Address (on cycles from the host side)
  - — C/A parity error on Host-initiated cycles, or
  - — Command parity error on Completion packets to the Intel® 3100 Chipset
- Host-Initiated Write Data
- Read Completion Data to the EHC

When any of these three errors are detected and the Parity Error Response bit is set in the USB 2.0 function, the EHC immediately sets the Host Error bit to ‘1’ and clears the Run/Stop bit to ‘0’. At most, one more packet completes on the high-speed USB ports after this occurs (this packet will not contain, or be based upon, the data containing the host error). When that packet is completed, the HCHalted bit is set to a ‘1’. Once the Host Error bit has been set, the Run/Stop bit can not be set by software until after the HCRESET is generated by software and completed by the EHC, which lasts for multiple milliseconds. The EHC will still accept host-initiated cycles as a target after the HCHalted bit has been set.

It is recommended that software reboot in the event of a parity error because the error could be an indication of other system hardware problems.

28.10 USB 2.0 Power Management

28.10.1 Pause Feature

This feature allows platforms to dynamically enter low-power states during brief periods when the system is idle (i.e., between keystrokes). This is useful for enabling power management features like Enhanced Intel SpeedStep Technology (EIST). The policies for entering these states typically are based on the recent history of system bus activity to incrementally enter deeper power management states. Normally, when the EHC is enabled, it regularly accesses main memory while traversing the DMA schedules looking for work to do; this activity is viewed by the power management software as a non-idle system, thus preventing the power managed states to be entered. Suspending all of the enabled ports can prevent the memory accesses from occurring, but there is an inherent latency overhead with entering and exiting the suspended state on the USB ports that makes this unacceptable for the purpose of dynamic power management. As a result, the EHCI software drivers are allowed to pause the EHC's DMA engines when it knows that the traffic patterns of the attached devices can afford the delay. The pause only prevents the EHC from generating memory accesses; the SOF packets continue to be generated on the USB ports (unlike the suspended state).

The expected sequence of events for the Pause Feature is:

1. When starting the DMA engines for the first time, the enable bits are set at the same time as, or after, the Run bit is set. However, the EHC should be capable of handling the Run bit set to 0 while one or both of the enable bits are 1; this may happen, for example, when the hardware halts the DMA due to an error. The enable bits may be set to 1 by different writes to the Command Register. The EHC takes the following actions when the enable bits are set by software:
   a. The corresponding Asynch/Periodic Schedule Status bit(s) is (are) immediately set to 1.
   b. If the Asynch Enable bit is set, the first queue head in the asynchronous schedule is immediately fetched (if the Bus Master Enable bit in Configuration space is set).
   c. If the Periodic Enable bit is set, then the periodic frame list entry is fetched (if the Bus Master Enable bit in Configuration space is set) on the next internal trigger point, which may be up to 1 ms later.
2. Before clearing a Schedule Enable bit, software reads the USB 2.0 Status register to make sure that the corresponding Schedule Status bit has been set.
3. When system software determines that it should pause the EHC schedule, one or both of the Schedule Enable bits are written to 0. When this happens, the EHC responds as follows:
   a. The schedule disables are handled independently. In other words, the asynchronous and periodic disables may take effect in any order and vary greatly in latency.
   b. If the Periodic Schedule Enable is cleared, up to two more periodic transactions may be seen on the USB ports. Reads associated with the periodic schedule cease when the first fetch for a new transaction would normally be initiated; any reads required to execute an already partially-fetched transaction will continue to be generated. Writes associated with the periodic schedule may continue until all pending transactions in the periodic DMA engine's transaction queue are completed. The Periodic Schedule Status bit is cleared when the memory reads have completed and the memory writes have been internally posted.

Note: Multiple high-bandwidth packets are considered one transaction.

a. If the Asynchronous Schedule Enable is cleared, up to two more asynchronous transactions may be seen on the USB ports. Reads associated with the asynchronous schedule cease when the first fetch for a new transaction would normally be initiated; any reads required to execute an already partially-fetched transaction will continue to be generated. Writes associated with the asynchronous schedule will continue until all pending transactions in the asynchronous DMA engine's transaction queue are completed. The Asynchronous Schedule Status bit is cleared when the memory reads have completed and the memory writes have been internally posted.

4. Before setting a Schedule Enable bit, software reads the USB 2.0 Status register to make sure that the corresponding Schedule Status bit is cleared.
5. When system software determines that it should reenable the EHC, one or both of the Schedule Enable are written to 1. When this happens, the EHC responds as described in the initial start-up case above.

The Intel® 3100 Chipset does *not* implement a similar pause mechanism in the classic host controllers, which conflicts with the recommendation in the *EHCI Specification*.

28.10.2 Suspend Feature

The *EHCI Specification*, Section 4.3 describes the details of Port Suspend and Resume.

28.10.3 ACPI Device States

The USB 2.0 function only supports the D0 and D3 PCI Power Management states. Notes regarding implementation of the Device States:

1. The EHC hardware does not inherently consume any more power when it is in the D0 state than it does in the D3 state. However, software is required to suspend or disable all ports prior to entering the D3 state such that the maximum power consumption is reduced.
2. In the D0 state, all implemented EHC features are enabled.
3. In the D3 state, accesses to the EHC memory-mapped I/O range will master abort. Since the Debug Port uses the same memory range, the Debug Port is only operational when the EHC is in the D0 state.
4. In the D3 state, the EHC interrupt must never assert for any reason. The internal PME# signal is used to signal wake events, etc.
5. When the Device PowerState field is written to D0 from D3, an internal reset is generated. See Section 28.4.3 for general rules on the effects of this reset.
6. Attempts to write any other value into the Device PowerState field other than 00b (D0 state) and 11b (D3 state) will complete normally without changing the current value in this field.

Software performs the following sequence to put the EHC in the D3 state:

1. Software selectively suspends any enabled EHC ports.
2. Software reads back the selectively suspended ports to make sure the EHC has completed the suspend request.
3. Software clears the Run bit.
4. Software reads back the USB 2.0 Status Register to verify that the EHC is halted.
5. Software reads and saves the contents of the PCI configuration registers for restoring the context after transitioning back to the D0 state.
6. Software writes the Device PowerState field to the D3 state.

28.10.4 ACPI System States

The EHC behavior as it relates to other power management states in the system is summarized in the following list:

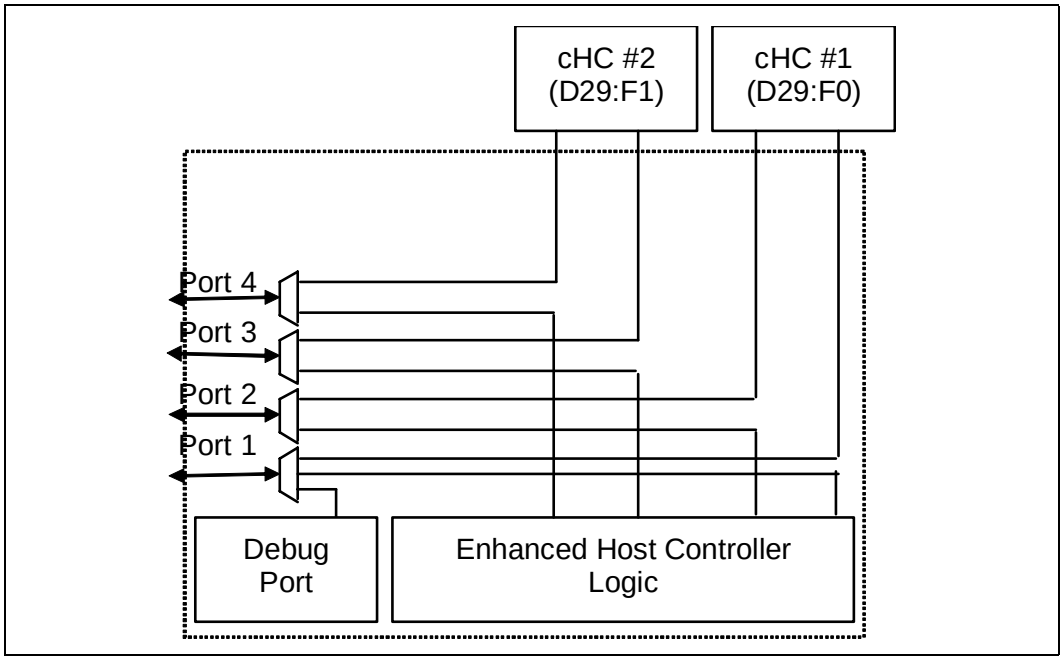
- The System is always in the S0 state when the EHC is in the D0 state. However, when the EHC is in the D3 state, the system may be in any power management state (including S0).
- When in D0, the Pause feature (Section 28.10.1, “Pause Feature”) enables dynamic CPU low-power states to be entered.
- The PLL in the EHC is disabled when entering the S3-Hot state (48 MHz clock stops), or the S3-Cold/S5 states (core power turns off).
- All core-well logic is reset in the S3/S5 states.

28.11 Interaction with Classic Host Controllers

The Enhanced Host Controller shares the four USB ports with two UHCI Classic Host Controllers (cHCs). The cHC at Device 29: Function 0 shares ports 1 and 2; the cHC at Device 29:Function 1 shares ports 3 and 4; with the EHC. There is very little interaction between the Enhanced and Classic controllers other than the muxing control that is provided as part of the EHC.

Figure 85 depicts the USB Port Connections at a conceptual level. The dashed rectangle indicates all of the logic that is part of the Enhanced Host Controller cluster.

Figure 85. USB Port Connections



28.11.1 Port-Routing Logic

Integrated into the EHC functionality is "port-routing logic," which performs the muxing between the classic and enhanced Host Controllers. The Intel® 3100 Chipset conceptually implements this logic as described in Section 4.2 of the *EHCI Specification, Rev. 1.0.* If a device is connected that is not capable of USB 2.0's high-speed signaling protocol or if the EHCI software drivers are not present as indicated by the Configured Flag, then the cHC owns the port. Owning the port means that the differential output is driven by the owner and the input stream is only visible to the owner. The HC that is not the owner of the port internally sees a disconnected port.

Note: The port-routing logic is the only block of logic that observes the physical (real) connect/disconnect information. The port status logic inside each of the host controllers observes the electrical (artificial) connect/disconnect information that is generated by the port-routing logic.

Only the differential signal pairs are muxed/demuxed between the classic and enhanced host controllers. The other USB functional signals are handled as follows:

- The Overcurrent inputs (OC#[3:0]) are directly routed to both controllers. An overcurrent event is recorded in both controllers' status registers.
- The Port Routing logic is implemented in the Suspend power well so that reenumeration and remapping of the USB ports is not required following entering and exiting a system sleep state in which the core power is turned off.
- The Intel® 3100 Chipset also allows the USB Debug Port traffic to be routed in and out of Port #0. When in this mode, the Enhanced Host Controller is the owner of Port #0.

28.11.2 Device Connects

The *EHCI Specification, Rev. 1.0.*describes the details of handling Device Connects in Section 4.2. There are four general scenarios that are summarized below.

1. Configure Flag = 0 and a Classic-only Device is connected

   In this case, the classic Host Controller is the owner of the port both before and after the connect occurs; the EHC (except for the port-routing logic) never sees the connect occur. The UHCI driver handles the connection and initialization process.

2. Configure Flag = 0 and an Enhanced-capable Device is connected

   In this case, the classic Host Controller is the owner of the port both before and after the connect occurs; the EHC (except for the port-routing logic) never sees the connect occur. The UHCI driver handles the connection and initialization process. Since the classic Host Controller does not perform the high-speed chirp handshake, the device operates in compatible mode.

3. Configure Flag = 1 and a Classic-only Device is connected

   In this case, the enhanced Host Controller is the owner of the port before the connect occurs. The EHCI driver checks the Line Status bits to determine if a low-speed device is connected. If so, then the Port Owner bit is written to a 1 and the UHCI driver handles the reset sequence. If a low-speed device is not detected through the Line Status bits, the EHCI driver handles the connection and performs the port reset. After the reset process completes, the EHC hardware has cleared (not set) the Port Enable bit in the EHC's PORTSC register. The EHCI driver then writes a 1 to the Port Owner bit in the same register, causing the classic Host Controller to see a connect event and the EHC to see an "electrical" disconnect event. The UHCI driver and hardware handle the connection and initialization process from that point on. The EHCI driver and hardware handle the perceived disconnect.

4. Configure Flag = 1 and an Enhanced-capable Device is connected

   In this case, the enhanced Host Controller is the owner of the port before, and remains the owner after, the connect occurs. The EHCI driver handles the connection and performs the port reset. After the reset process completes, the EHC hardware has set the Port Enable bit in the EHC's PORTSC register. The port is functional at this point. The classic Host Controller continues to see an unconnected port.

28.11.3 Device Disconnects

The *EHCI Specification, Rev. 1.0.*describes the details of handling Device Connects in Section 4.2. There are three general scenarios that are summarized below.

1. Configure Flag = 0 and the device is disconnected

   In this case, the classic Host Controller is the owner of the port both before and after the disconnect occurs; the EHC (except for the port-routing logic) never sees a device attached. The UHCI driver handles disconnection process.

2. Configure Flag = 1 and a Classic Device is disconnected

   In this case, the classic Host Controller is the owner of the port before the disconnect occurs. The disconnect is reported by the classic Host Controller and serviced by the associated UHCI driver. The port-routing logic in the EHC cluster forces the Port Owner bit to 0, indicating that the EHC owns the unconnected port.

3. Configure Flag = 1 and an Enhanced Device is disconnected

   In this case, the enhanced Host Controller is the owner of the port before, and remains the owner after, the disconnect occurs. The EHCI hardware and driver handle the disconnection process. The classic Host Controller never sees a device attached.

28.11.4 Effect of Resets on Port-Routing Logic

As mentioned above, the Port Routing logic is implemented in the Suspend power well so that reenumeration and remapping of the USB ports is not required following entering and exiting a system sleep state in which the core power is turned off.

Table 868. Effect of Resets on Port-Routing Logic

Reset Event	Effect on Configure Flag	Effect on Port Owner Bits
Suspend Well Reset	cleared (0)	set (1)
Core Well Reset	no effect	no effect
D3-to-D0 Reset	no effect	no effect
HCRESET	cleared (0)	set (1)

28.12 USB 2.0 Legacy Keyboard Operation

The Intel® 3100 Chipset must support the possibility of a keyboard downstream from either a USB1 (low-speed or full-speed) or a USB 2.0 (high-speed) port. See Section 25.12, "USB Legacy Keyboard Operation" for the description of the legacy keyboard support.

The EHC provides the basic ability to generate SMIs on an interrupt event, along with more sophisticated control of the generation of SMIs, as documented in Section 28.2.1.27, "Offset 68 - 6Bh: ULSEC – USB 2.0 Legacy Support Extended Capability Register".

28.13 USB 2.0 Based Debug Port

The Intel® 3100 Chipset supports the elimination of the legacy COM ports by providing the ability for new debugger software to interact with devices on a USB 2.0 port.

High-level restrictions and features:

- Must be operational before USB 2.0 drivers are loaded.
  - Must work even when the port is disabled.
  - Must work even though non-configured port is default-routed to the classic controller.

Note: The Debug Port cannot be used to debug an issue that requires a classic USB device on Port #0 using the UHCI drivers.

- Must allow normal system USB 2.0 traffic in a system that may only have one USB port.
- Debug Port device (DPD) must be High-Speed capable and connect to a High-Speed port on the Intel® 3100 Chipset systems.
- Debug Port FIFO must always make forward progress (a bad status on USB is simply presented back to software).

The Debug Port FIFO is only given one USB access per microframe.

28.13.1 USB 2.0 Based Debug Port Overview

The Debug port facilitates OS and device driver debug. It allows the software to communicate with an external console using a USB 2.0 connection. Because the interface to this link does not go through the normal USB 2.0 stack, it allows communication with the external console during cases where the OS is not loaded, the USB 2.0 software is broken, or where the USB 2.0 software is being debugged.

Specific features of this implementation of a debug port are:

- Only works with an external USB 2.0 debug device (console)
- Implemented for a specific port on the host controller
- Operational anytime the port is not suspended AND the host controller is in D0 power state.
- Capability is interrupted when port is driving USB RESET

28.13.2 Debug Port Registers

The Debug port's registers are located in the same memory range as the standard EHCI registers, which are defined by the Base Address Register (BAR). The base offset for these registers (A0h) is declared in the Debug Port Base Offset Capability Register at Configuration offset 5Ah. The specific EHCI port that supports this debug capability is indicated by a four-bit field (bits 20-23) in the HCSPARAMS register of an EHCI controller.

The map of the Debug Port registers is shown in Table 869. Each register is defined individually in the register tables.

Behavioral Rules:

1. All of these registers are implemented in the core well and reset by EHC HCRESET, EHC D3-to-D0 state transition, and PLTRST#.
2. The hardware associated with this register provides no checks to ensure that software programs the interface correctly. How the hardware behaves when programmed illegally is undefined.

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 869. Debug Port Registers Summary Table

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
A0h	A3h	CNTL_STS	Control/Status Register	00000000h	RW,RWC,RO,WO
A4h	A4h	USBPID	USB PIDs	0000000h	Bits 31:16 are Read-Only, Bits 15:00 are Read/Write
A8h	AFh	DATABUF	Data Buffer (Bytes 7:0)	00000000 00000000h	RW
B0h	B0h	CONFIG	Configuration Register	00007F01	RW

28.13.2.1 Offset A0 - A3h: CNTL_STS – Control/Status Register

Software must do Read-Modify-Write operations to this register to preserve the contents of bits not being modified. This include Reserved bits.

In order to preserve the usage of RESERVED bits in the future, software must always write the same value read from the bit until it is defined. Reserved bits will always return 0 when read.

Table 870. Offset A0 - A3h: CNTL_STS – Control/Status Register (Sheet 1 of 2)

I/O Address: A0 -A3h ***Size:*** 32 bit

Default Value: 00000000h

Bits	Name	Description	Reset Value	Access
31	Reserved	Reserved	0	
30	OWNER_CNT	0 = Ownership of the debug port is NOT forced to the EHCI controller (Default) 1 = Ownership of the debug port is forced to the EHCI controller (i.e., immediately taken away from the companion Classic USB Host controller). If the port was already owned by the EHCI controller, then setting this bit has no effect. This bit overrides all of the ownership-related bits in the standard EHCI registers. The value in this bit does not affect the value reported in the PORTSC Port Owner bit.	0	RW
29	Reserved	Reserved	0	
28	ENABLED_CNT	0 = Software can clear this by writing a 0 to it. The hardware clears this bit for the same conditions where the Port Enable/Disable Change bit (in the PORTSC register) is set. (Default) 1 = Debug port is enabled for operation. Software can directly set this bit if the port is already enabled in the associated PORTSC register (this is enforced by the hardware).	0	RW
27	Reserved	Reserved	0	
26:17	Reserved	Reserved	0	
16	DONE_STS	0 = Request Not complete. 1 = Set by hardware to indicate that the request is complete. Writing a 1 to this bit will clear it if it is set. Writing a 0 to this bit has no effect. Reset default = 0.	0	RWC
15:12	LINK_ID_STS	This field identifies the link interface. It is hardwired to 0h to indicate that it is a USB Debug Port.	0	RO
11	Reserved	Reserved	0	
10	IN_USE_CNT	Set by software to indicate that the port is in use. Cleared by software to indicate that the port is free and may be used by other software. This bit is cleared after reset. (This bit has no effect on hardware.)	0	RW
09:07	EXCEPTION_STS	This field indicates the exception when the ERROR_GOOD#_STS bit is set. This field must be ignored if the ERROR_GOOD#_STS bit is 0. 000 No Error. ***Note:*** This must not be seen, since this field must only be checked if there is an error. 001 Transaction error: indicates the USB 2.0 transaction had an error (CRC, bad PID, timeout, etc.) 010 Hardware error. Request was attempted (or in progress) when port was suspended or reset. All others are reserved. Reset default = 000b	000b	RO

Table 870. Offset A0 - A3h: CNTL_STS – Control/Status Register (Sheet 2 of 2)

I/O Address: A0 -A3h		*Size:* 32 bit		
Default Value: 00000000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
06	ERROR_ GOOD#_STS	0 = The hardware clears this bit to 0 upon the proper completion of a read or write. 1 = The hardware sets this bit to indicate that an error has occurred. Details on the nature of the error are provided in the Exception field. Reset default = 0.	0	RO
05	GO_CNT	Software sets this bit to cause the hardware to perform a read or write request. Writing a 0 to this bit has no effect. Writing a 1 to this bit when it is already set may result in undefined behavior. When set, the hardware clears this bit when the hardware sets the DONE_STS bit. Reset default = 0.	0	RW
04	WRITE_READ#_ CNT:	Software sets this bit to indicate that the current request is a write. Software clears this bit to indicate that the current request is a read. Reset default = 0.	0	RW
03:00	DATA_LEN_CNT:	This field is used to indicate the size of the data to be transferred. For write operations, this field is set by software to indicate to the hardware how many bytes of data in Data Buffer are to be transferred to the console. A value of 0h indicates that a zero-length packet must be sent. A value of 1-8 indicates 1-8 bytes are to be transferred. Values 9-Fh are illegal and how hardware behaves if used is undefined. For read operations, this field is set by hardware to indicate to software how many bytes in Data Buffer are valid in response to a read operation. A value of 0h indicates that a zero length packet was returned and the state of Data Buffer is not defined. A value of 1-8 indicates 1-8 bytes were received. Hardware is not allowed to return values 9-Fh. The transferring of data always starts with byte 0 in the data area and moves toward byte 7 until the transfer size is reached. Reset default = 0h.	0	RW

28.13.2.2 Offset A4h: USBPID — USB PIDs Register

This Dword register is used to communicate PID information between the USB debug driver and the USB debug port. The debug port uses some of these fields to generate USB packets, and uses other fields to return PID information to the USB debug driver.

Table 871. Offset A4h: USBPID – USB PIDs Register (Sheet 1 of 2)

I/O Address: A4h		*Size:* 32 bit		
Default Value: *0000000h*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:24	Reserved	Reserved	0	

Table 871. Offset A4h: USBPID – USB PIDs Register (Sheet 2 of 2)

I/O Address: A4h		Size: 32 bit		
Default Value: 0000000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
23:16	RECEIVED_PID_STS	The hardware updates this field with the received PID for transactions in either direction. When the controller is writing data, this field is updated with the handshake PID that is received from the device. When the host controller is reading data, this field is updated with the data packet PID (if the device sent data), or the handshake PID (if the device NAKs the request). This field is valid when the hardware clears the GO_DONE#_CNT bit.	0	RO
15:08	SEND_PID_CNT	The hardware sends this PID to begin the data packet when sending data to USB (i.e., WRITE_READ#_CNT is asserted). Software will typically set this field to either DATA0 or DATA1 PID values.	0	RW
07:00	TOKEN_PID_CNT	The hardware sends this PID as the Token PID for each USB transaction. Software will typically set this field to either IN, OUT or SETUP PID values.	0	RW

28.13.2.3 Offset A8 - AFh: DATABUF – Data Buffer Bytes 7:0

Note: This register can be accessed as eight separate 8-bit registers or two separate 32-bit registers.

Table 872. Offset A8 - AFh: DATABUF – Data Buffer Bytes 7:0

I/O Address: A8 - AFh		Size: 64 bit		
Default Value: 0000000000000000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
63:00	DATABUFFER	These are the 8 bytes of the data buffer. Bits 7:0 correspond to least significant byte (byte 0). Bits 63:56 correspond to the most significant byte (byte 7). The bytes in the Data Buffer must be written with data before software initiates a write request. For a read request, the Data Buffer contains valid data when DONE_STS is set by the hardware, ERROR_GOOD#_STS is cleared by the hardware, and the DATA_LENGTH_CNT field indicates the number of bytes that are valid.	0h	RW

28.13.2.4 Offset B0h: CONFIG – Configuration Register

Table 873. Offset B0h: CONFIG – Configuration Register

I/O Address: B0h		Size: 32 bit		
Default Value: 00007F01				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:15	Reserved	Reserved	0	
14:08	USB_ADDRESS_CNF	7-bit field that identifies the USB device address used by the controller for all Token PID generation. This is a RW field that is set to 7Fh after reset.	7Fh	RW
07:04	Reserved	Reserved	0	
03:00	USB_ENDPOINT_CNF	This 4-bit field identifies the endpoint used by the controller for all Token PID generation. This is a RW field that is set to 01h after reset.	1h	RW

28.13.3 USB 2.0 Based Debug Port Theory of Operation

There are two operational modes for the USB debug port:

1. Mode 1 is when the Enhanced USB Host Controller is in a disabled state from the viewpoint of a standard EHCI driver (i.e., Host Controller's *Run/Stop* bit is 0). In Mode 1, the Debug Port controller is required to generate 'keepalive' packets less than 2 milliseconds apart to keep the attached debug device from suspending. The keepalive packet must be a standalone 32-bit SYNC field.
2. Mode 2 is when the host controller is running (i.e., Host controller's *Run/Stop#* bit is 1). In Mode 2, the normal transmission of SOF packets (or SYNC keepalives if the port is functionally disabled) will keep the debug device from suspending.

28.13.3.1 Behavioral Rules

1. In both modes 1 and 2, the Debug Port controller must check for software requested debug transactions at least every 125 microseconds.
2. If the debug port is enabled by the debug driver, and the standard host controller driver resets the USB port, USB debug transactions are held off for the duration of the reset and until after the first SOF is sent.
3. If the standard host controller driver suspends the USB port, then USB debug transactions are held off for the duration of the suspend/resume sequence and until after the first SOF is sent.
4. The ENABLED_CNT bit in the debug register space is independent of the similar port control bit in the associated Port Status and Control register.

Table 874 shows the debug port behavior related to the state of bits in the debug registers as well as bits in the associated Port Status and Control register.

Table 874. Debug Port Behavior

Debug bits		EHCI bits			Debug port behavior
OWNER_CNT	ENABLED_CNT	Port Enable	Run/ Stop#	Suspend	
0	X	X	X	X	Debug port is not being used. Normal operation.
1	0	X	X	X	Debug port is not being used. Normal operation.
1	1	0	0	X	Debug port in Mode 1. SYNC keepalives sent plus debug traffic
1	1	0	1	X	Debug port in Mode 2. SYNC keepalives or SOF packets may be sent plus debug traffic. The Intel® 3100 Chipset generates SYNC keepalives, not SOF packets. No other normal traffic is sent out this port, because the port is not enabled.
1	1	1	0	0	Illegal. Host controller driver must never put the controller into this state (enabled, not running and not suspended).
1	1	1	0	1	Port is suspended. No debug traffic sent.
1	1	1	1	0	Debug port in Mode 2. Debug traffic is interspersed with normal traffic.
1	1	1	1	1	Port is suspended. No debug traffic sent.

28.13.3.2 OUT Transactions

An OUT Transaction sends data to the debug device. It can occur only when the following are true:

- The debug port is enabled
- The debug software sets the GO_CNT bit
- The WRITE_READ#_CNT bit is set

The sequence of the transaction is:

1. Software sets the appropriate values in the following bits:

   USB_ADDRESS_CNF
   USB_ENDPOINT_CNF
   DATA_BUFFER[63:0]
   TOKEN_PID_CNT[7:0]
   SEND_PID_CNT[15:8]
   DATA_LEN_CNT
   WRITE_READ#_CNT(note: this will always be 1 for OUT transactions)
   GO_CNT (Note: this will always be 1 to initiate the transaction)

2. The debug port controller sends a token packet consisting of:

   A. SYNC
   B. TOKEN_PID_CNT field
   C. USB_ADDRESS_CNT field
   D. USB_ENDPOINT_CNT field
   E. 5-bit CRC field

3. After sending the token packet, the debug port controller sends a data packet consisting of:

   F. SYNC
   G. SEND_PID_CNT field
   H. The number of data bytes indicated in DATA_LEN_CNT from the DATA_BUFFER
   I. 16-bit CRC

Note: A DATA_LEN_CNT value of zero is valid in which case no data bytes would be included in the packet.

4. After sending the data packet, the controller waits for a handshake response from the debug device.

- If a handshake is received, the debug port controller:

  J. Places the received PID in the RECEIVED_PID_STS field
  K. Resets the ERROR_GOOD#_STS bit
  L. Sets the DONE_STS bit

- If no handshake PID is received, the debug port controller:

  J. Sets the EXCEPTION_STS field to 001b
  K. Sets the ERROR_GOOD#_STS bit
  L. Sets the DONE_STS bit

28.13.3.3 IN Transactions

An IN transaction receives data from the debug device. It can occur only when the following are true:

- The debug port is enabled

- The debug software sets the GO_CNT bit
- The WRITE_READ#_CNT bit is reset

The sequence of the transaction is:

1. Software sets the appropriate values in the following bits:

   USB_ADDRESS_CNF
   USB_ENDPOINT_CNF
   TOKEN_PID_CNT[7:0]
   DATA_LEN_CNT
   WRITE_READ#_CNT(note: this will always be 0 for IN transactions)
   GO_CNT (note: this will always be 1 to initiate the transaction)

   — The debug port controller sends a token packet consisting of:

   — SYNC

   — TOKEN_PID_CNT field

   — USB_ADDRESS_CNT field

   — USB_ENDPOINT_CNT field

   — 5-bit CRC field.

2. After sending the token packet, the debug port controller waits for a response from the debug device.

   If a response is received:
   The received PID is placed into the RECEIVED_PID_STS field

   — Any subsequent bytes are placed into the DATA_BUFFER

   — The DATA_LEN_CNT field is updated to show the number of bytes that were received after the PID.

3. If a valid packet was received from the device that was one byte in length (indicating it was a handshake packet), then the debug port controller:

   — Resets the ERROR_GOOD#_STS bit

   — Sets the DONE_STS bit

4. If a valid packet was received from the device that was more than one byte in length (indicating it was a data packet), then the debug port controller:

   — Transmits an ACK handshake packet

   — Resets the ERROR_GOOD#_STS bit

   — Sets the DONE_STS bit

5. If no valid packet is received, then the debug port controller:

   — Sets the EXCEPTION_STS field to 001b

   — Sets the ERROR_GOOD#_STS bit,

   — Sets the DONE_STS bit.

28.13.3.4 Debug Software

28.13.3.4.1 Enabling the Debug Port

There are two mutually exclusive conditions that debug software must address as part of its startup processing:

- The EHCI has been initialized by system software.
- The EHCI has not been initialized by system software.

Debug software can determine the current 'initialized' state of the EHCI by examining the *Configure Flag* in the EHCI USB 2.0 Command Register. If this flag is set, then system software has initialized the EHCI. Otherwise, the EHCI must not be considered initialized. Debug software will initialize the debug port registers depending on the state of the EHCI. However, before this can be accomplished, debug software must determine which root USB port is designated as the debug port.

28.13.3.4.2 Determining the Debug Port

Debug software can easily determine which USB root port has been designated as the debug port by examining bits 20:23 of the EHCI Host Controller Structural Parameters register. This 4-bit field represents the numeric value assigned to the debug port (i.e., 0001 == port 0, 0010 == port 1, 0011 == port 2 …… 1111 == port 14). This value is 0001 (port 0).

28.13.3.4.3 Debug Software Startup with Non-Initialized EHCI

Debug software can attempt to use the debug port if, after setting the *OWNER_CNT* bit, the *Current Connect Status* bit in the appropriate (See Section 28.13.3.4.2, "Determining the Debug Port") PORTSC register is set. If the *Current Connect Status* bit is not set, then debug software may choose to terminate, or it may choose to wait until a device is connected.

If a device is connected to the port, then debug software must reset/enable the port. Debug software does this by setting and then clearing the *Port Reset* bit in the PORTSC register. Software must set the *Run/Stop* bit in the EHCI Command Register before clearing the *Port Reset* bit in order to complete the reset and to enable the port. To guarantee a successful reset, debug software must also keep the *Port Reset* bit set for at least 50 ms. Due to possible delays, this bit may not change to zero immediately; reset is complete when this bit reads as zero. Software must not continue until this bit reads zero.

If a high-speed device is attached, the EHCI will automatically set the *Port Enabled/Disabled* bit in the PORTSC register and the debug software can proceed. Debug software must set the *ENABLED_CNT* bit in the Debug Port Control/Status register, and then reset (clear) the *Port Enabled/Disabled* bit in the PORTSC register and the *Run/Stop* bit in the EHCI Command Register. The EHCI bits are cleared in order to present the proper default idle conditions to the EHCI driver as it loads.

28.13.3.4.4 Debug Software Startup with Initialized EHCI

Debug software can attempt to use the debug port if the *Current Connect Status* bit in the appropriate (See Section 28.13.3.4.2, "Determining the Debug Port") PORTSC register is set. If the *Current Connect Status* bit is not set, then debug software may terminate or it may wait until a device is connected.

If a device is connected, then debug software must set the *OWNER_CNT* bit and then the *ENABLED_CNT* bit in the Debug Port Control/Status register.

28.13.3.4.5 Determining Debug Peripheral Presence

After enabling the debug port functionality, debug software can determine if a debug peripheral is attached by attempting to send data to the debug peripheral. If all attempts result in an error (*Exception* bits in the Debug Port Control/Status register indicates a Transaction Error), then the attached device is not a debug peripheral. If the debug port peripheral is not present, then debug software may terminate or it may wait until a debug peripheral is connected.

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

29.0 Device 31, Function 2: SATA Host Controller

29.1 Overview

All of the SATA configuration registers are in the core well. None of the registers can be locked. All registers that are not shown must be treated as reserved.

Table 875. Configuration Register Summary Table

Start	End	Functionality Described
00h	3Fh	PCI Header
40h	5Fh	SFF-8038i Configuration (legacy bus master IDE)
70h	7Fh	PCI Power Management Capability Pointer
80h	8Fh	Message Signaled Interrupt Capability Pointer
90h	FFh	Additional Configuration

29.1.1 PCI Header

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 876. PCI Header Registers Summary Table (Sheet 1 of 2)

Offset		Symbol	Name	Default	Access
Start	End				
00h	03h	ID	Identifiers Register	See Desc 8086	RO
04h	05h	CMD	Command Register	0000h	RO, RW
06h	07h	STS	Device Status Register	02B0h	RO, RWC
08h	08h	RID	Revision ID Register	See Desc	RO
09h	09h	PI	Programming Interface Register	01h	RO
0Ah	0Bh	CC	Class Codes 00 Register	See register description	RO, RWOnce
0Dh	0Dh	MLT	Master Latency Timer Register	00h	RO
10h	10h	PCMDBA	Primary Command Block Base Address Register	00000001h	RO, RW
14h	17h	PCTLBA	Primary Control Block Base Address Register	00000001h	RO, RW
18h	1Bh	SCMDBA	Secondary Command Block Base Address Register	00000001h	RO, RW
1Ch	1Fh	SCTLBA	Secondary Control Block Base Address Register	00000001h	RO, RW

Table 876. PCI Header Registers Summary Table (Sheet 2 of 2)

Offset		Symbol	Name	Default	Access
Start	End				
20h	23h	LBAR	Legacy Bus Master IDE Base Address Register	00000001h	RO, RW
24h	27h	ABAR	AHCI Base Address Register	00000000h	RO, RW
2Ch	2Fh	SS	Sub System Identifiers Register	00000000h	RO, RW
34h	34h	CAP	Capabilities Pointer Register	70h	RO
3Ch	3Dh	INTR	Interrupt Information Register	See register description	RO, RW

29.1.1.1 Offset 00 - 03h: ID – Identifiers Register

Table 877. Offset 00 - 03h: ID – Identifiers Register

Device: 31 ***Function:*** 2
Offset: 00 - 03h ***Size:*** 32 bit
Default Value: *See Desc 8086* ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:16	DID	**Device ID:** The specific value is dependent on MAP.SMS (see Table 912, Bits7:6) and MAP.MV (Table 912, Bits1:0). See Table 881 in Section 29.1.1.5. for the reset value.	See Desc	RO
15:00	VID	**Vendor ID:** 16-bit field that indicates the company vendor as Intel.	8086h	RO

29.1.1.2 Offset 04 - 05h: CMD – Command Register

Table 878. Offset 04 - 05h: CMD – Command Register (Sheet 1 of 2)

Device: 31 ***Function:*** 2
Offset: 04 - 05h ***Size:*** 16 bit
Default Value: *0000h* ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:11	Reserved	Reserved	0	
10	ID	**Interrupt Disable:** This disables pin-based INTx# interrupts. This bit has no effect on MSI operation. 0 = Internal INTx# messages are generated if there is an interrupt and MSI is not enabled. 1 = Internal INTx# messages will not be generated.	0	RW
09	FBE	**Fast Back-to-Back Enable:** Reserved	0	
08	SEE	**SERR# Enable:** Reserved. The SATA Controller never generates an SERR#.	0	
07	WCC	**Wait Cycle Enable:** Reserved	0	
06	PEE	**Parity Error Response Enable:** 0 = Disabled. SATA controller will not generate PERR# when a data parity error is detected. 1 = Enabled. SATA controller will generate PERR# when a data parity error is detected.	0	RW
05	VGA	**VGA Palette Snooping Enable:** Reserved	0	
04	MWIE	**Memory Write and Invalidate Enable:** Reserved	0	
03	SCE	**Special Cycle Enable:** Reserved	0	

Table 878. Offset 04 - 05h: CMD – Command Register (Sheet 2 of 2)

Device: 31	**Function:** 2
Offset: 04 - 05h	**Size:** 16 bit
Default Value: 0000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
02	BME	**Bus Master Enable:** 0 = Disable SATA Controller to the act as bus master 1 = Enable SATA Controller to act as a master for data transfer This bit does not impact the generation of completions for split transaction commands.	0	RW
01	MSE	**Memory Space Enable:** Controls access to the SATA Controller's target memory space (for AHCI). When the MAP.MV register is programmed for a combined mode, this register becomes read-only. ***Note:*** Hardware does not clear this bit to 0 when switching from non-combined to combined mode. Software is responsible for clearing this bit before switching to combined mode.	0	RW,RO
00	IOSE	**I/O Space Enable:** Controls access to the SATA Controller's target I/O space. 0 = Disables access to the Legacy IDE ports (both Primary and Secondary) as well as the Bus Master I/O registers. 1 = Enable. The Base Address register for the Bus Master registers must be programmed before this bit is set.	0	RW

29.1.1.3 Offset 06 - 07h: STS – Device Status Register

Note: For the writable bits, software must write a 1 to clear bits that are set. Writing a 0 to the bit has no effect.

Table 879. Offset 06 - 07h: STS – Device Status Register (Sheet 1 of 2)

Device: 31	**Function:** 2
Offset: 06 - 07h	**Size:** 16 bit
Default Value: 02B0h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
15	DPE	**Detected Parity Error:** 0 = No parity error detected by SATA controller. 1 = SATA controller detects a parity error on its interface.	0	RWC
14	SSE	**Signaled System Error:** Reserved. The SATA Controller will never generate an SERR#.	0	
13	RMA	**Received Master-Abort Status:** 0 = Master abort not generated. 1 = Set when the SATA Controller receives a master abort to a cycle it generated.	0	RWC
12	RTA	**Received Target-Abort Status:** 0 = Target abort not generated. 1 = Set when the SATA Controller receives a target abort to a cycle it generated.	0	RWC
11	STA	**Signaled Target-Abort Status:** Reserved. The SATA Controller will never generate a target abort.	0	
10:09	DEVT	**DEVSEL# Timing Status:** Controls the device select time for the SATA Controller's PCI interface.	01	RO

Table 879. Offset 06 - 07h: STS – Device Status Register (Sheet 2 of 2)

Device: 31	***Function:*** 2
Offset: 06 - 07h	***Size:*** 16 bit
Default Value: 02B0h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
08	DPD	**Master Data Parity Error Detected:** 0 = No parity error detected or parity line asserted. 1 = Set when the SATA Controller, as a master, either detects a parity error or sees the parity error line asserted, and the parity error response bit (bit 6 of the command register) is set. This bit can only be set on read completions received from the IICH where there is a parity error.	0	RWC
07	FBBC	**Fast Back-to-Back Capable:** Reserved.	1	
06	Reserved	Reserved	0	
05	Reserved	**66 MHz Capable:** Reserved	1	
04	CL	**Capabilities List:** Indicates the presence of a capabilities list. The minimum requirement for the capabilities list must be PCI power management for the SATA Controller. 0 = Capabilities list is not present 1 = Capabilities list is present	1	RO
03	IS	**Interrupt Status:** Reflects the state of INTx# messages. 0 = This bit is a 0 after the interrupt is cleared (independent of the state of CMD.ID). 1 = This bit is set when the interrupt is to be asserted.	0	RO
02:00	Reserved	Reserved	0	

29.1.1.4 Offset 08h: RID – Revision ID Register

Table 880. Offset 08h: RID – Revision ID Register

Device: 31	***Function:*** 2
Offset: 08h	***Size:*** 8 bit
Default Value: *See Desc*	***Power Well:***

Bits	Name	Description	Reset Value	Access
07:00	RID	**Revision ID:** Indicates stepping of the host controller hardware. The value reported in this register depends on the value written to the Revision ID in Device 31, Function 0, Offset 08h.	See Desc	RO

29.1.1.5 Offset 09h: PI – Programming Interface Register

Table 881 illustrates the dependencies upon the SATA HW register programming and the resultant values for: Device ID, Device Class Code and Device Programming Interface (PI).

Table 881. Decode of PI, CC and DID Combinations

Input Parameters From Port Mapping Register (Table 912)	Resulting values		
MAP.SMS (See bits 7:6 Table 912 for full bit descriptions): 00–IDE 01–AHCI 10– 11-Reserved	CC.SCC (Bits 7:0) (See Table 884, for full descriptions)	PI (See below for full descriptions)	Device ID (DID) (See Table 877, for full descriptions)
00b	01h (IDE)	8Ah (RW)	2680h
01b	06h (AHCI) (SATA)	01h (RO)	2681h

Table 882. Offset 09h: PI - When Sub Class Code Register (D31:F2:Offset 0Ah, CC.SCC) = '01h'

Device: 31 ***Function:*** 2

Offset: 09h ***Size:*** 8 bit

Default Value: *1000X0X0* ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07	BMO	**Bus Master Operation:** Indicates the SATA Controller supports bus master operation.	1	RO
06:04	Reserved	Reserved	0	
03	SNC	**Secondary Mode Native Capable:** Indicates that the secondary controller supports both legacy and native modes. 0 = Secondary controller only supports legacy mode. 1 = Secondary controller supports both legacy and native modes. When MAP.MV (Table 912, D31:F2 Offset 90, bits 1:0) is any value other than 00b, this bit reports as a '0'. When MAP.MV is 00b, this bit reports as a '1'.	See description	RO

Table 882. Offset 09h: PI - When Sub Class Code Register (D31:F2:Offset 0Ah, CC.SCC) = '01h'

Device: 31	***Function:*** 2
Offset: 09h	***Size:*** 8 bit
Default Value: *1000X0X0*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
02	SNE	**Secondary Mode Native Enable:** Determines the mode that the secondary channel is operating in. 0 = Secondary controller operating in legacy (compatibility) mode. 1 = Secondary controller operating in native PCI mode. If this bit is set by software, then the PNE bit (bit 0 of this register) must also be set by software. While in theory these bits can be programmed separately, such a configuration is not supported by today's software and is not supported by this hardware. When MAP.MV (Table 912, D31:F2 Offset 90:bits 1:0) is any value other than 00b, this bit is read-only. When MAP.MV is 00b, this bit is read/write. ***Note:***	0	RW/RO
01	PNC	**Primary Mode Native Capable:** Indicates that the primary controller supports both legacy and native modes. 0 = Primary controller only supports legacy mode. 1 = Primary controller supports both legacy and native modes. When MAP.MV (Table 912, D31:F2 Offset 90:bits 1:0) is any value other than 00, this bit reports as a '0'. When MAP.MV is '00', this bit reports as a '1'.	See description	RO
00	PNE	**Primary Mode Native Enable:** Determines the mode that the primary channel is operating in. 0 = Primary controller operating in legacy (compatibility) mode. 1 = Primary controller operating in native PCI mode. If this bit is set by software, then the SNE bit (bit 2 of this register) must also be set by software. While in theory these bits can be programmed separately, such a configuration is not supported by today's software and is not supported by this hardware. When MAP.MV (Table 912, D31:F2 Offset 90:bits 1:0) is any value other than 00, this bit is read-only. When MAP.MV is '00', this bit is read/write. ***Note:***	0	RW/RO

Table 883. Offset 09h: PI - When Sub Class Code Register (D31:F2:Offset 0Ah, CC.SCC) = '06h'

Device: 31	***Function:*** 2
Offset: 09h	***Size:*** 8 bit
Default Value: *01h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	IF	**Interface:** Indicates the SATA Controller is AHCI 1.0 compliant.	01h	RO

29.1.1.6 Offset 0A - 0Bh: CC – Class Code Register

Table 884. Offset 0A - 0Bh: CC – Class Code Register

Device: 31	***Function:*** 2
Offset: 0A - 0Bh	***Size:*** 16 bit
Default Value: *See register description*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:08	BCC	**Base Class Code: 01h** Indicates that this is a mass storage device. All other values are undefined.	01h	RO
07:00	SCC	**Sub Class Code:** The value reported in this field is dependent on MAP.SMS (Table 912, D31:F2 Offset 90:bits 7:6) and MAP.MV (Table 912, D31:F2 Offset 90:bits 1:0). See Table 881 in Section 29.1.1.5 for the reset value.	See description	RO

29.1.1.7 Offset 0Dh: MLT – Master Latency Timer Register

Table 885. Offset 0Dh: MLT – Master Latency Timer Register

Device: 31	***Function:*** 2
Offset: 0Dh	***Size:*** 8 bit
Default Value: *00h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	MLT	**Master Latency Timer:** This register has no meaning for the SATA controller .	00h	RO

29.1.1.8 Offset 10h: PCMDBA – Primary Command Block Base Address Register

This 8-byte I/O space is used in Native Mode for the Primary Controller's Control Block.

Table 886. Offset 10h: PCMDBA – Primary Command Block Base Address Register

Device: 31	***Function:*** 2
Offset: 10h	***Size:*** 32 bit
Default Value: 00000001h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0	
15:03	BAR	**Base Address:** Base address of the I/O space (eight consecutive I/O locations).	0	RW
02:01	Reserved	Reserved	0	
00	RTE	**Resource Type Indicator:** 0 = No request for I/O space. 1 = Indicates a request for I/O space.	1	RO

29.1.1.9 Offset 14 - 17h: PCTLBA – Primary Control Block Base Address Register

This 4-byte I/O space is used in Native Mode for the Primary Controller's Control Block.

Table 887. Offset 14 - 17h: PCTLBA – Primary Control Block Base Address Register

Device: 31	***Function:*** 2
Offset: 14 - 17h	***Size:*** 32 bit
Default Value: 00000001h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0	
15:02	BAR	**Base Address:** Base address of the I/O space (four consecutive I/O locations).	0	RW
01	Reserved	Reserved	0	
00	RTE	**Resource Type Indicator:** 0 = No request for I/O space. 1 = Indicates a request for I/O space.	1	RO

29.1.1.10 Offset 18 - 1Bh: SCMDBA – Secondary Command Block Base Address Register

This 8-byte I/O space is used in Native Mode for the Secondary Controller's Command Block.

Table 888. Offset 18 - 1Bh: SCMDBA – Secondary Command Block Base Address Register

Device: 31	***Function:*** 2
Offset: 18 - 1Bh	***Size:*** 32 bit
Default Value: 00000001h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0	
15:03	BAR	**Base Address:** Base address of the I/O space (eight consecutive I/O locations).	0	RW
02:01	Reserved	Reserved	0	
00	RTE	**Resource Type Indicator:** 0 = No request for I/O space. 1 = Indicates a request for I/O space.	1	RO

29.1.1.11 Offset 1C - 1Fh: SCTLBA – Secondary Control Block Base Address Register

This 4-byte I/O space is used in Native Mode for the Secondary Controller's Control Block.

Table 889. Offset 1C - 1Fh: SCTLBA – Secondary Control Block Base Address Register

Device: 31	**Function:** 2
Offset: 1C - 1Fh	**Size:** 32 bit
Default Value: 00000001h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0	
15:02	BAR	**Base Address:** Base address of the I/O space (four consecutive I/O locations).	0	RW
01	Reserved	Reserved	0	
00	RTE	**Resource Type Indicator:** 0 = No request for I/O space. 1 = Indicates a request for I/O space.	1	RO

29.1.1.12 Offset 20 - 23h: LBAR – Legacy Bus Master Base Address Register

This BAR is used to allocate I/O space for the SFF-8038i mode of operation and AHCI Index/Data pair. When the sub-class code is SATA and PI as AHCI, this BAR can be used for indirect AHCI register accesses.

The size of I/O space is 32 bytes when the sub-class code is SATA and PI as AHCI. Otherwise, it is only 16 bytes. This means the index and the data register pair are not available when the sub-class code is not SATA with AHCI PI.

The following is the register definition if the sub-class code is SATA with AHCI PI. (Base address + 16 bytes) is the address of the Index register (Indirect address register) and the data register is located at (Base address + 16 bytes + 4 bytes).

Table 890. Offset 20 - 23h: LBAR – Legacy Bus Master Base Address Register when SCC is SATA with AHCI PI

Device: 31	**Function:** 2
Offset: 20 - 23h	**Size:** 32 bit
Default Value: 00000001h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0	
15:05	BA	**Base Address:** Base address of the I/O space (32 consecutive I/O locations).	0	RW
04:01	Reserved	Reserved	0	
00	RTE	**Resource Type Indicator:** 0 = No request for I/O space. 1 = Indicates a request for I/O space.	1	RO

The following is the register definition when the sub-class code is not SATA with AHCI PI. In this case the 16-byte I/O space is used in SFF-8038i mode.

Table 891. Offset 20 - 23h: LBAR – Legacy Bus Master Base Address Register when SCC is not SATA with AHCI PI

Device: 31	***Function:*** 2
Offset: 20h	***Size:*** 32 bit
Default Value: 00000001h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0	
15:04	BA	**Base Address:** Base address of the I/O space (16 consecutive I/O locations).	0	RW
03:01	Reserved	Reserved	0	
00	RTE	**Resource Type Indicator:** 0 = No request for I/O space. 1 = Indicates a request for I/O space.	1	RO

29.1.1.13 Offset 24 - 27h: ABAR – AHCI Base Address Register

This register allocates space for the memory registers defined in Section 29.3.

Table 892. Offset 24 - 27h: ABAR – AHCI Base Address Register

Device: 31	***Function:*** 2
Offset: 24 - 27h	***Size:*** 32 bit
Default Value: 00000000h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:10	BA	**Base Address:** Base address of register memory space (aligned to 1 KByte)	0	RW
09:04	Reserved	Reserved	0	
03	PF	**Prefetchable:** 0 = Indicates that this range is prefetchable. 1 = Indicates that this range is not prefetchable.	0	RO
02:01	TP	**Type:** Indicates that this range can be mapped anywhere in 32-bit address space.	00	RO
00	RTE	**Resource Type Indicator:** 0 = Indicates no request for register memory space. 1 = Indicates a request for register memory space.	0	RO

29.1.1.14 Offset 2C - 2Fh: SS – Sub System Identifiers Register

This register is initialized to logic 0 by the assertion of PLTRST#. This register can be written only once after PLTRST# deassertion.

Table 893. Offset 2C - 2Fh: SS – Sub System Identifiers Register

Device: 31	***Function:*** 2
Offset: 2C - 2Fh	***Size:*** 32 bit
Default Value: *00000000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:16	SSID	**Subsystem ID:** This is written by BIOS. No hardware action taken on this value.	0000h	RWO
15:00	SSVID	**Subsystem Vendor ID:** This is written by BIOS. No hardware action taken on this value.	0000h	RWO

29.1.1.15 Offset 34h: CAP – Capabilities Pointer Register

Table 894. Offset 34h: CAP – Capabilities Pointer Register

Device: 31	***Function:*** 2
Offset: 34h	***Size:*** 8 bit
Default Value: 70h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	CP	**Capability Pointer:** Indicates that the first capability pointer offset is offset 70h (the Power Management Capability).	70h	RO

29.1.1.16 Offset 3C - 3Dh: INTR – Interrupt Information Register

Table 895. Offset 3C - 3Dh: INTR – Interrupt Information Register

Device: 31	***Function:*** 2
Offset: 3C - 3Dh	***Size:*** 16 bit
Default Value: See register description.	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:08	IPIN	**Interrupt Pin:** This reflects the value of D31IP.SIP in configuration space.	See description	RO
07:00	ILINE	**Interrupt Line:** Software written value to indicate which interrupt line (vector) the interrupt is connected to. No hardware action is taken on this register.	00h	RW

29.1.2 Additional SFF-8038i Configuration Registers

The following registers are necessary to implement as read/write bits in order to maintain software functionality for the SFF-8038i mode of operation. They have no bearing on Serial ATA operation unless otherwise indicated, but are necessary to be read/write for legacy software capability.

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 896. Register Summary: Additional SFF-8038i Configuration Registers

Offset		Symbol	Name	Default	Type
Start	End				
40h	41h	PTIM	Primary IDE Timing Register	*0000h*	RW
42h	43h	STIM	Secondary IDE Timing Register	0000h	RW
44h	44h	D1TIM	Slave IDE Timing Register	00h	RW
48h	48h	SYNCC	Synchronous DMA Control Register	00h	RO, RW
4Ah	4Bh	SYNCTIM	Synchronous DMA Timing Register	0000h	RO, RW
54h	57h	IIOC	IDE I/O Configuration Register	*00000000h*	RO, RW

29.1.3 Register Details

29.1.3.1 Offset 40 - 41h: PTIM – Primary Timing Register

This controls the timings driven on the parallel cable.

Table 897. Offset 40 - 41h: PTIM – Primary Timing Register (Sheet 1 of 2)

Device: 31 ***Function:*** 2
Offset: 40 - 41h ***Size:*** 16 bit
Default Value: *0000h* ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15	DE	**Decode Enable:** 0 = Disable. 1 = Enables the SATA Controller to decode the associated Command Blocks (1F0–1F7h for primary, 170–177h for secondary or their native BAR equivalents) and Control Block (3F6h for primary and 376h for secondary or their native BAR equivalents). This bit affects the IDE decode ranges for both legacy and native-mode decoding. This bit still has functionality in SATA – if this bit is not set ('1'), the port that is mapped to this range will not be decoded.	0	RW
14	D1STE	**Device 1 Separate Timing Enable:** 0 = Use bits 13:12, 09:08 for both drive 0 and drive 1. 1 = Use bits 13:12, 09:08 for drive 0, and use the Slave IDE Timing register at offset 44h for drive 1.	0	RW
13:12	ISP	**IORDY Sample Point:** Determines the number of 33 MHz clocks between IDE IOR#/IOW# assertion and the first IORDY sample point. 00 5 clocks 01 4 clocks 10 3 clocks 11 Reserved	00	RW
11:10	Reserved	Reserved	00	
09:08	RCT	**Recovery Time:** The setting of these bits determines the minimum number of 33 MHz clocks between the last IORDY sample point and the IOR#/IOW# strobe of the next cycle. 00 4 clocks 01 3 clocks 10 2 clocks 11 1 clock	00	RW

Table 897. Offset 40 - 41h: PTIM – Primary Timing Register (Sheet 2 of 2)

Device: 31	**Function:** 2
Offset: 40 - 41h	**Size:** 16 bit
Default Value: 0000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
07	DTE1	**Device 1 DMA Timing Enable:** 0 = Disable. 1 = Enable the fast timing mode for DMA transfers only for this drive. PIO transfers to the IDE data port will run in compatible timing.	0	RW
06	PPE1	**Device 1 Prefetch / Posting Enable:** 0 = Disable. 1 = Enable Prefetch and posting to the IDE data port for this drive.	0	RW
05	IE1	**Device 1 IORDY Sample Point Enable:** 0 = Disable IORDY sampling for this drive. 1 = Enable IORDY sampling for this drive.	0	RW
04	TIM1	**Device 1 Fast Timing Bank:** 0 = Accesses to the data port will use compatible timings for this drive. 1 = When this bit =1 and bit 14 = 0, accesses to the data port will use bits 13:12 for the IORDY sample point, and bits 09:08 for the recovery time. When this bit = 1 and bit 14 = 1, accesses to the data port will use the IORDY sample point and recover time specified in the slave IDE timing register.	0	RW
03	DTE0	**Device 0 DMA Timing Enable:** 0 = The fast timing mode is disabled for DMA transfers only for this drive. 1 = The fast timing mode is enabled for DMA transfers only for this drive. PIO transfers to the IDE data port will run in compatible timing.	0	RW
02	PPE0	**Device 0 Prefetch / Posting Enable:** 0 = Prefetch and posting to the IDE data port is disabled for this drive. 1 = Prefetch and posting to the IDE data port is enabled for this drive.	0	RW
01	IE0	**Device 0 IORDY Sample Point Enable:** 0 = IORDY sampling is disabled for this drive. 1 = IORDY sampling is enabled for this drive.	0	RW
00	TIM0	**Device 0 Fast Timing Bank:** 0 = Accesses to the data port will use compatible timings for this drive. 1 = Accesses to the data port will use bits 13:12 for the IORDY sample point, and bits 09:08 for the recovery time.	0	RW

29.1.3.2 Offset 42 - 43h: STIM – Secondary Timing Register

See the above register description for Primary IDE Timing.

29.1.3.3 Offset 44h: D1TIM – Device 1 IDE Timing Register

The values in this register are only valid if the “Separate Timing Enable” bit in, Table 897, “Offset 40 - 41h: PTIM – Primary Timing Register” on page 899, is set (‘1’) in the timing registers. Bits 07:04 are used by device 1 on secondary if bit 14 of offset 42h is set, and bits 03:00 are used by device 1 on primary if bit 14 of offset 40h is set.

Table 898. Offset 44h: D1TIM – Device 1 IDE Timing Register

Device: 31	**Function:** 2
Offset: 44h	**Size:** 8 bit
Default Value: 00h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
07:06	SISP1	**Secondary Device 1 IORDY Sample Point:** Determines the number of 33 MHz clocks between IDE IOR#/IOW# assertion and the first IORDY sample point. 00 5 clocks 01 4 clocks 10 3 clocks 11 Reserved	00	RW
05:04	SRCT1	**Secondary Device 1 Recovery Time:** Determines the minimum number of 33 MHz clocks between the last IORDY sample point and the IOR#/IOW# strobe of the next cycle. 00 4 clocks 01 3 clocks 10 2 clocks 11 1 clocks	00	RW
03:02	PISP1	**Primary Device 1 IORDY Sample Point:** Same as bits 07:06, except for the primary device.	00	RW
01:00	PRCT1	**Primary Device 1 Recovery Time:** Same as bits 05:04, except for the primary device.	00	RW

29.1.3.4 Offset 48h: SYNCC – Synchronous DMA Control Register

Table 899. Offset 48h: SYNCC – Synchronous DMA Control Register

Device: 31	**Function:** 2
Offset: 48h	**Size:** 8 bit
Default Value: 00h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	0	
03	SDAE1	**Secondary Device 1 ATAxx Enable:** 0 = Disables ATA33/66/100/133 timing modes for the secondary slave device. 1 = Enables ATA33/66/100/133 timing modes for the secondary slave device.	0	RW
02	SDAE0	**Secondary Device 0 ATAxx Enable:** 0 = Disables ATA33/66/100/133 timing modes for the secondary master device. 1 = Enables ATA33/66/100/133 timing modes for the secondary master device.	0	RW
01	PDAE1	**Primary Device 1 ATAxx Enable:** 0 = Disables ATA33/66/100/133 timing modes for the primary slave device. 1 = Enables ATA33/66/100/133 timing modes for the primary slave device.	0	RW
00	PDAE0	**Primary Device 0 ATAxx Enable:** 0 = Disables ATA33/66/100/133 timing modes for the primary master device. 1 = Enables ATA33/66/100/133 timing modes for the primary master device.	0	RW

29.1.3.5 Offset 4A - 4Bh: SYNCTIM – Synchronous DMA Timing Register

The CT and RP values for the ATA_FAST (100 MBytes/s and 133 MBytes/s modes) are based on 133 MHz clock. The CT and RP values for the 66 MHz and 33 MHz modes are based on either a 66 MHz or 33 MHz clock.

Table 900. Offset 4A - 4Bh: SYNCTIM – Synchronous DMA Timing Register

Device: 31	***Function:*** 2
Offset: 4A - 4Bh	***Size:*** 16 bit
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:14	Reserved	Reserved	0	
13:12	SCT1	**Secondary Device 1 Cycle Time:** The setting of these bits determines the minimum write strobe cycle time (CT) and DMARDY#-to-STOP (RP) time. Bits: SBC[3] = 0 CT/RP; SBC[3] = 1 CT/RP; FSBCE[3] = 1 CT/RP 00: 4/6; Reserved; Reserved 01: 3/5; 3/8; 3/16 10: 2/4; 2/8; Reserved 11: Reserved; Reserved; Reserved	00	RW
11:10	Reserved	Reserved	0	
09:08	SCT0	**Secondary Device 0 Cycle Time:** Same definition as bits 13:12, except for device 0.	00	RW
07:06	Reserved	Reserved	0	
05:04	PCT1	**Primary Device 1 Cycle Time:** Same definition as bits 13:12, except for primary device 1.	00	RW
03:02	Reserved	Reserved	0	
01:00	PCT0	**Primary Device 0 Cycle Time:** Same definition as bits 13:12, except for primary device 0.	00	RW

29.1.3.6 Offset 54 - 57h: IIOC – IDE I/O Configuration Register

Table 901. Offset 54 - 57h: IIOC – IDE I/O Configuration Register (Sheet 1 of 2)

Device: 31	***Function:*** 2
Offset: 54 - 57h	***Size:*** 32 bit
Default Value: *00000000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:24	Reserved	Reserved	0	
23:20	SP2	**Scratchpad:** No hardware action taken on these bits.	0h	RW
19:18	SSM	**Secondary Signals Mode:** Controls the secondary signals for swap bays. 00 Normal (Enabled) 01 Tri-state (Disabled) 10 Drive low (Disabled) 11 Reserved	00	RW

Table 901. Offset 54 - 57h: IIOC – IDE I/O Configuration Register (Sheet 2 of 2)

Device: 31	***Function:*** 2
Offset: 54 - 57h	***Size:*** 32 bit
Default Value: *00000000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
17:16	PSM	**Primary Signals Mode:** Controls the primary signals for swap bays. 00 Normal (Enabled) 01 Tri-state (Disabled) 10 Drive low (Disabled) 11 Reserved	00	RW
15:12	FSBCE	**Fast Synchronous Base Clock Enable:** 0 = Disables fast ATA modes. 1 = Enables fast ATA modes. This overrides the state of the SCB[3:0] bits in this register.	0h	RW
11:10	Reserved	Reserved	0	
09:08	Reserved	**Reserved:** These were the secondary and primary command posting enable bits. These do not exist when parallel ATA is mapped as part of serial ATA, because the command posting BAR has been removed.	0	RO
07:04	SP1	**Scratchpad:** No hardware action taken on these bits.	0h	RW
03:00	SBC	**Synchronous Base Clock:** Clock used to determine CT and RP timings for synchronous DMA timings. 0 = 33 MHz clock used, 1 = 66 MHz clock used. Bit 3 controls the secondary slave device Bit 2 controls the secondary master device Bit 1 controls the primary slave device Bit 0 controls the primary master device	0h	RW

29.1.4 PCI Power Management Capabilities

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 902. Register Summary: PCI Power Management Capabilities Registers

Offset		Symbol	Name	Default	Access
Start	End				
70h	71h	PID	PCI Power Management Capability ID Register	*0001h*	RO
72h	73h	PC	PCI Power Management Capabilities Register	*4002h*	RO
74h	75h	PMCS	PCI Power Management Control and Status Register	*0000h*	RO, RW, RWC

29.1.4.1 Offset 70 - 71h: PID – PCI Power Management Capability ID Register

Table 903. Offset 70 - 71h: PID – PCI Power Management Capability ID Register

Device: 31	***Function:*** 2
Offset: 70 - 71h	***Size:*** 16 bit
Default Value: *0001h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:08	NEXT	**Next Capability:** When CC.SCC is 01h, this field will be 00h indicating this is the last item in the list. When CC.SCC is not 01h, this field will be A8h pointing to the Serial ATA Capability structure.	00h or A8h	RO
07:00	CID	**Cap ID:** Indicates that this pointer is a PCI power management.	01h	RO

29.1.4.2 Offset 72 - 73h: PC – PCI Power Management Capabilities Register

Table 904. Offset 72 - 73h: PC – PCI Power Management Capabilities Register

Device: 31	***Function:*** 2
Offset: 72 - 73h	***Size:*** 16 bit
Default Value: *4002h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:11	PME_Support	Indicates PME# can be generated from the $D3_{HOT}$ state in the SATA controller	01000	RO
10	D2_Support	The D2 state is not supported.	0	RO
09	D1_Support	The D1 state is not supported.	0	RO
08:06	Aux_Current	PME# from $D3_{COLD}$ state is not supported, therefore this field is 000b.	000	RO
05	DSI	**Device Specific Initialization:**. 0 = Indicates device-specific initialization is required 1 = Indicates that no device-specific initialization is required.	0	RO
04	Reserved	Reserved	0	
03	PMEC	**PME Clock:** 0 = Indicates PCI clock is required to generate PME#. 1 = Indicates that PCI clock is not required to generate PME#.	0	RO
02:00	VS	**Version:** Indicates support for Revision 1.1 of the *PCI Power Management Specification.*	010	RO

29.1.4.3 Offset 74 - 75h: PMCS – PCI Power Management Control And Status Register

Table 905. Offset 74 - 75h: PMCS – PCI Power Management Control And Status Register

Device: 31	***Function:*** 2
Offset: 74 - 75h	***Size:*** 16 bit
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15	PMES	**PME Status:** 0 = Set to 0. 1 = This bit is set to logic 1, when PME event is to be requested, and if PMEE is set to a logic 1 then a PME# is generated.	0	RWC
14:09	Reserved	**Reserved**	0	
08	PMEE	**PME Enable:** 0 = Set to 0. 1 = When set to a logic 1, the SATA controller generates PME# from $D3_{HOT}$ on a wake event.	0	RW
07:02	Reserved	Reserved	0	
01:00	PS	**Power State:** This field is used both to determine the current power state of the SATA Controller and to set a new power state. The values are: 00 D0 state 11 $D3_{HOT}$ state When in the $D3_{HOT}$ state, the controller's configuration space is available, but the I/O and memory spaces are not. Additionally, interrupts are blocked. If software attempts to write a '10' or '01' to these bits, the write is ignored.	00	RW

29.1.5 Message Signaled Interrupt Capability

Note: Even though the MSI Capability ID and MSI registers are fully implemented, MSI functionality is not supported. The PCI Capability Pointer structure bypasses the MSI capability.

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 906. Register Summary: Message Signaled Interrupt Registers

Offset		Symbol	Name	Default	Access
Start	End				
80h	81h	MSIID	Message Signaled Interrupt Identifiers Register	*7005h*	RO
82h	83h	MSIC	Message Signaled Interrupt Message Control Register	*0000h*	RO, RW
84h	87h	MSIA	Message Signaled Interrupt Message Address Register	*00000000h*	RO, RW
88h	89h	MSID	Message Signaled Interrupt Message Data Register	*0000h*	RW

29.1.5.1 Offset 80 - 81h: MSIID – Message Signaled Interrupt Identifiers Register

Table 907. Offset 80 - 81h: MSIID – Message Signaled Interrupt Identifiers Register

Device: 31	***Function:*** 2
Offset: 80 - 81h	***Size:*** 16 bit
Default Value: *7005h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:08	NEXT	**Next Pointer:** Indicates the next item in the list is the PCI power management pointer.	70h	RO
07:00	CID	**Capability ID:** Capabilities ID indicates MSI.	05h	RO

29.1.5.2 Offset 82 - 83h: MSIC – Message Signaled Interrupt Message Control Register

Table 908. Offset 82 - 83h: MSIC – Message Signaled Interrupt Message Control Register

Device: 31	***Function:*** 2
Offset: 82 - 83h	***Size:*** 16 bit
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:08	Reserved	Reserved	0	
07	C64	**64 Bit Address Capable:** 0 = Not capable of generating a 32-bit message. 1 = Capable of generating a 32-bit message only.	0	RO
06:04	MME	**Multiple Message Enable:** These bits are read/write for software compatibility, but only one message is ever sent by the SATA controller.	000	RW
03:01	MMC	**Multiple Message Capable:** Only one message is required.	000	RO
00	MSIE	**MSI Enable:** 0 = Disabled. 1 = MSI is enabled and traditional interrupt pins are not used to generate interrupts.	0	RW

29.1.5.3 Offset 84 - 87h: MSIA – Message Signaled Interrupt Message Address Register

Table 909. Offset 84 - 87h: MSIA – Message Signaled Interrupt Message Address Register

Device: 31	***Function:*** 2
Offset: 84 - 87h	***Size:*** 32 bit
Default Value: *00000000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:02	ADDR	**Address:** Lower 32-bits of the system-specified message address, always Dword aligned.	0	RW
01:00	Reserved	Reserved	00	

29.1.5.4 Offset 88 - 89h: MSID – Message Signaled Interrupt Message Data Register

Table 910. Offset 88h: MSID – Message Signaled Interrupt Message Data Register

Device: 31	***Function:*** 2
Offset: 88 - 89h	***Size:*** 16 bit
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:00	DATA	**Data:** This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the lower word of the data bus of the MSI memory write transaction.	0	RW

29.1.6 Additional Configuration Registers

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 911. Additional Configuration Registers Summary

Offset		Symbol	Name	Default	Type
Start	End				
90h	90h	MAP	Port Mapping Register	*00h*	RO, RW
91h	93h	PCS	Port Control and Status Register	*000000*	RO, RW, RWC
A0h	A0h	SIRI	SATA Indexed Register Index	00h	RO, RW
A4h	A7h	STRD	SATA Indexed Register Data	XXXXXXXXh	RW
A8h	ABh	ACR0	AHCI Capability Register 0	00100012h	RO
ACh	AFh	ACR1	AHCI Capability Register 1	00000048h	RO
C0h	C0h	ATC	APM Trapping Control Register	00h	RO, RW
C4h	C4h	ATS	APM Trapping Status Register	00h	RO, RWC
D0h	D3h	SP	Scratch Pad Register	00000000h	RW
E0h	E3h	BFCS	BIST FIS Control/Status Register	00000000h	RO, RW, RWC
E4h	E7h	BFTD1	BIST FIS Transmit Data, DW1 Register	00000000h	RW
E8h	EBh	BFTD2	BIST FIS Transmit Data, DW2 Register	00000000h	RW
F8h	F8h	MID	Manufacturer's ID Register	00010F80h	RO

29.1.6.1 Offset 90h: MAP – Port Mapping Register

Table 912. Offset 90h: MAP – Port Mapping Register

Device: 31	***Function:*** 2
Offset: 90h	***Size:*** 8 bit
Default Value: *00h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:06	SMS	**SATA Mode Select:** Software programs these bits to control the mode in which the SATA Host Bus Adaptor (HBA) should operate: 00 IDE mode 01 AHCI Mode 10 Reserved 11 Reserved ***Notes:*** AHCI mode may only be selected when MV=0 and AHCI is enabled. Programming these bits with values that are illegal will result in indeterministic behavior by the hardware.	0	RW
05:02	Reserved	Reserved	0	
01:00	MV	**Map Value:** The value in the bits below indicate the address range the SATA ports responds to **Bits** / **Mode** / **Primary: Master** / **Primary: Slave** / **Secondary: Master** / **Secondary: Slave** 00 / SATA / Port 0 / Port 2 / Port 1 / Port 3 01,10,11 / Reserved	00	RO

29.1.6.2 Offset 91 - 93h: PCS – Port Control and Status Register

Table 913. Offset 91 - 93h: PCS – Port Control and Status Register (Sheet 1 of 3)

Device: 31	***Function:*** 2
Offset: 91 - 93h	***Size:*** 24 bit
Default Value: *000000*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
23	Reserved	Reserved	0	
22:21	Reserved	Reserved	0	
20	ORM	**OOB Retry Mode:** 0 = The SATA controller will not retry after an OOB failure. 1 = The SATA controller will continue to retry after an OOB failure until successful (infinite retry).	0	RW
19:16	Reserved	Reserved	0	

Table 913. Offset 91 - 93h: PCS – Port Control and Status Register (Sheet 2 of 3)

Device: 31	***Function:*** 2
Offset: 91 - 93h	***Size:*** 24 bit
Default Value: *000000*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15	P3P	**Port 3 Present:** The status of this bit may change at any time. This bit is cleared when the port is disabled via P3E. This bit is not cleared upon surprise removal of a device 0 = No device detected 1 = The SATA controller has detected the presence of a device on port 3.	0	RO
14	P2P	**Port 2 Present:** The status of this bit may change at any time. This bit is cleared when the port is disabled via P2E. This bit is not cleared upon surprise removal of a device 0 = No device detected 1 = The SATA controller has detected the presence of a device on port 2.	0	RO
13	P1P	**Port 1 Present:** The status of this bit may change at any time. This bit is cleared when the port is disabled via P1E. This bit is not cleared upon surprise removal of a device 0 = No device detected 1 = The SATA controller has detected the presence of a device on port 1.	0	RO
12	P0P	**Port 0 Present:** The status of this bit may change at any time. This bit is cleared when the port is disabled via P0E. This bit is not cleared upon surprise removal of a device 0 = No device detected 1 = The SATA controller has detected the presence of a device on port 0.	0	RO
11	P3E	**Port 3 Enabled:** 0 = Disabled. The port is in the 'off' state and cannot detect any devices. 1 = Enabled. The port can transition between the on, partial, and slumber states and can detect devices. ***Note:*** This bit takes precedence over P3CMD.SUD (offset ABAR+298h:bit 1. See Table 892.)	0	RW
10	P2E	**Port 2 Enabled:** 0 = Disabled. The port is in the 'off' state and cannot detect any devices. 1 = Enabled. The port can transition between the on, partial, and slumber states and can detect devices. ***Note:*** This bit takes precedence over P2CMD.SUD (offset ABAR+218h:bit 1. See Table 892.)	0	RW
09	P1E	**Port 1 Enabled:** 0 = Disabled. The port is in the 'off' state and cannot detect any devices. 1 = Enabled. The port can transition between the on, partial, and slumber states and can detect devices. ***Note:*** This bit takes precedence over P1CMD.SUD (offset ABAR+198h:bit 1. See Table 892.)	0	RW
08	P0E	Port 0 Enabled: 0 = Disabled. The port is in the 'off' state and cannot detect any devices. 1 = Enabled. The port can transition between the on, partial, and slumber states and can detect devices. ***Note:*** This bit takes precedence over P0CMD.SUD (offset ABAR+118h:bit 1. See Table 892.)	0	RW
07:04	Reserved	Reserved	0000	

Table 913. Offset 91 - 93h: PCS – Port Control and Status Register (Sheet 3 of 3)

Device: 31	**Function:** 2
Offset: 91 - 93h	**Size:** 24 bit
Default Value: *000000*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
03	P5P	**Port 5 Present:** This bit is cleared when the port is disabled via P5E. This bit is not cleared upon surprise removal of a device. 0 = No device detected. 1 = The SATA controller has detected the presence of device on port 5. It may change at any time.	0	RO
02	P4P	**Port 4 Present:** This bit is cleared when the port is disabled via P4E. This bit is not cleared upon surprise removal of a device. 0 = No device detected. 1 = The SATA controller has detected the presence of device on port 4. It may change at any time.	0	RO
01	P5E	**Port 5 Enabled:** This bit takes precedence over P5CMD.SUD. 0 = Disabled. The port is in the 'off' state and cannot detect any devices. 1 = Enabled. The port can transition between the on, partial, and slumber states and can detect devices.	0	RW
00	P4E	**Port 4 Enabled:** This bit takes precedence over P4CMD.SUD. 0 = Disabled. The port is in the 'off' state and cannot detect any devices. 1 = Enabled. The port can transition between the on, partial, and slumber states and can detect devices.	0	RW

29.1.6.3 Offset A0h: SIRI – SATA Indexed Register Index

Table 914. Offset A0h: SIRI – SATA Indexed Register Index

Device: 31	**Function:** 2
Offset: A0h	**Size:** 8 bit
Default Value: *00h*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
07:02	IDX	**Index:** 6-bit index pointer into the SATA Indexed Register space. Data is written into and read from the STRD register. This points to a dWord register. The byte enables on the STRD register affect what is written.	00h	RW
01:00	Reserved	Reserved	00	

29.1.6.4 Offset A4 - A7h: STRD – SATA Indexed Register Data

Table 915. Offset A4 - A7h: STRD – SATA Indexed Register Data

Device: 31	***Function:*** 2
Offset: A4 - A7h	***Size:*** 32 bit
Default Value: *XXXXXXXXh*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:00	DTA	**Data:** 32-bit data value that is written to the register pointed to by SIRI, or read from the register pointed to by SIRI.	*XXXXXXXX*h	RW

29.1.6.5 Offset A8 - ABh: ACR0 – AHCI Capability Register 0

The following is the register definition when the sub-class code is SATA with AHCI PI.

Table 916. Offset A8 - ABh: ACR0 – AHCI Capability Register 0

Device: 31	***Function:*** 2
Offset: A8 - ABh	***Size:*** 32 bit
Default Value: *00100012h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:24	Reserved	Reserved	00h	
23:20	MAJR	**MAJOR Revision:** Specifies the major revision level to which the SATA HBA has been implemented.	1h	RO
19:16	MINR	**MINOR Revision:** Specifies the minor revision level to which the SATA HBA has been implemented.	0h	RO
15:08	NEXT	**Next Capability Pointer:** Pints to the next capability structure. 00h indicates this is the last capability pointer.	00h	RO
07:00	CID	**Capability ID:** This value has been assigned by the PCI SIG to designate a generic index/data pair mechanism.	12h	RO

29.1.6.6 Offset AC - AFh: ACR1 – AHCI Capability Register 1

The following is the register definition when the sub-class code is SATA with AHCI PI.

29.1.6.7 Offset C0h: ATC – APM Trapping Control Register

Table 917. Offset AC - AFh: ACR1 – AHCI Capability Register 1

Device: 31	**Function:** 2
Offset: AC - AFh	**Size:** 32 bit
Default Value: *00000048h*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0000h	
15:04	BAROFF	**BAR Offset:** Indicates the offset into the BAR where the Index/Data pair are located (in Dword granularity). The Index and Data registers are located at offset 10h within the I/O space defined by LBAR. A value of 004h indicates offset 10h.	004h	RO
03:00	BARLOC	**BAR Location:** Indicates the offset of the BAR containing the Index/ Data pair (in Dword granularity). The index and Data I/O registers reside within the space defined by LBAR in the SATA controller. A value of 8h indicates offset 20h which is LBAR. LBAR is the BAR location for Index/Data registers.	8h	RO

Table 918. Offset C0h: ATC – APM Trapping Control Register

Device: 31	**Function:** 2
Offset: C0h	**Size:** 8 bit
Default Value: *00h*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	0	
03	SST	**Secondary Slave Trap:** 0 = Disable. 1 = Enables trapping and SMI# assertion on legacy I/O accesses to 170h-177h and 376h. The active device on the secondary interface must be device 1 for the trap and/or SMI# to occur.	0	RW
02	SPT	**Secondary Master Trap:** 0 = Disable. 1 = Enables trapping and SMI# assertion on legacy I/O accesses to 170h-177h and 376h. The active device on the secondary interface must be device 0 for the trap and/or SMI# to occur.	0	RW
01	PST	**Primary Slave Trap:** 0 = Disable. 1 = Enables trapping and SMI# assertion on legacy I/O accesses to 1F0h-1F7h and 3F6h. The active device on the primary interface must be device 1 for the trap and/or SMI# to occur.	0	RW
00	PMT	**Primary Master Trap:** 0 = Disable. 1 = Enables trapping and SMI# assertion on legacy I/O accesses to 1F0h-1F7h and 3F6h. The active device on the primary interface must be device 0 for the trap and/or SMI# to occur.	0	RW

29.1.6.8 Offset C4h: ATS – ATM Trapping Status Register

Table 919. Offset C4h: ATS – ATM Trapping Status Register

Device: 31	***Function:*** 2
Offset: C4h	***Size:*** 8 bit
Default Value: *00h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	0	
03	SST	**Secondary Slave Trap:** 0 = Indicates no trap occurred to the secondary slave device. 1 = Indicates that a trap occurred to the secondary slave device.	0	RWC
02	SPT	**Secondary Master Trap:** 0 = Indicates no trap occurred to the secondary master device. 1 = Indicates that a trap occurred to the secondary master device.	0	RWC
01	PST	**Primary Slave Trap:** 0 = Indicates no trap occurred to the primary slave device. 1 = Indicates that a trap occurred to the primary slave device.	0	RWC
00	PMT	**Primary Master Trap:** 0 = Indicates no trap occurred to the primary master device. 1 = Indicates that a trap occurred to the primary master device.	0	RWC

29.1.6.9 Offset D0 - D3h: SP – Scratch Pad Register

Table 920. Offset D0 - D3h: SP – Scratch Pad Register

Device: 31	***Function:*** 2
Offset: *D0 - D3h*	***Size:*** 32 bit
Default Value: *00000000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:00	DT	**Data:** This is a read/write register that is available for software to use. No hardware action is taken on this register.	0	RW

29.1.6.10 Offset E0 - E3h: BFCS – BIST FIS Control/Status Register

Table 921. Offset E0 - E3h: BFCS – BIST FIS Control/Status Register (Sheet 1 of 2)

Device: 31	***Function:*** 2
Offset: E0 - E3h	***Size:*** 32 bit
Default Value: *00000000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0	
15	P5BFI	**Port 5 BIST FIS Initiate:** 0 = Disable. 1 = When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 5, using the parameters specified in this register and BFTD1 and BFTD2. The BIST FIS will only be initiated if a device is present and not in the partial or slumber states. Software must clear this bit before disabling port 5. After a BIST FIS is successfully completed, software must disable and re-enable PCS.P5E prior to attempting additional BIST FISes or to return the SATA controller to a normal operational mode. If the BIST FIS fails, as indicated by BFF in this register, software can clear then set this bit to initiate another BIST FIS.	0	RW
14	P4BFI	**Port 4 BIST FIS Initiate:** 0 = Disable. 1 = When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 4, using the parameters specified in this register and BFTD1 and BFTD2. The BIST FIS will only be initiated if a device is present and not in the partial or slumber states. Software must clear this bit before disabling port 4. After a BIST FIS is successfully completed, software must disable and re-enable PCS.P4E prior to attempting additional BIST FISes or to return the SATA controller to a normal operational mode. If the BIST FIS fails, as indicated by BFF in this register, software can clear then set this bit to initiate another BIST FIS.	0	RW
13	P3BFI	**Port 3 BIST FIS Initiate:** 0 = Disable. 1 = When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 3, using the parameters specified in this register and BFTD1 and BFTD2. The BIST FIS will only be initiated if a device is present and not in the partial or slumber states. Software must clear this bit before disabling port 3. After a BIST FIS is successfully completed, software must disable and reenable the PCS.P3E prior to attempting additional BIST FISes or to return the SATA controller to a normal operational mode. If the BIST FIS fails, as indicated by BFF in this register, software can clear, then set, this bit to initiate another BIST FIS.	0	RW
12	P2BFI	**Port 2 BIST FIS Initiate:** 0 = Disable. 1 = When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 2, using the parameters specified in this register and BFTD1 and BFTD2. The BIST FIS will only be initiated if a device is present and not in the partial or slumber states. Software must clear this bit before disabling port 2. After a BIST FIS is successfully completed, software must disable and reenable PCS.P2E prior to attempting additional BIST FISes or to return the SATA controller to a normal operational mode. If the BIST FIS fails, as indicated by BFF in this register, software can clear, then set, this bit to initiate another BIST FIS.	0	RW

Table 921. Offset E0 - E3h: BFCS – BIST FIS Control/Status Register (Sheet 2 of 2)

Device: 31	**Function:** 2
Offset: E0 - E3h	**Size:** 32 bit
Default Value: 00000000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
11	BFS	**BIST FIS Successful:** 0 = Software clears this bit by writing a 1 to it. 1 = This bit is set any time a BIST FIS transmitted by SATA Controller receives an R_OK completion status from the device. ***Note:*** This bit must be cleared by software prior to initiating a BIST FIS.	0	RWC
10	BFF	**BIST FIS Failed:** 0 = Software clears this bit by writing a 1 to it. 1 = This bit is set any time a BIST FIS transmitted by SATA Controller receives an R_ERR completion status from the device. ***Note:*** This bit must be cleared by software prior to initiating a BIST FIS.	0	RWC
09	P1BFI	**Port 1 BIST FIS Initiate:** 0 = Disable. 1 = When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 1, using the parameters specified in this register and BFTD1 and BFTD2. The BIST FIS will only be initiated if a device is present and not in the partial or slumber states. Software must clear this bit before disabling port 1. After a BIST FIS is successfully completed, software must disable and reenable PCS.P1E prior to attempting additional BIST FISes or to return the SATA controller to a normal operational mode. If the BIST FIS fails, as indicated by BFF in this register, software can clear then set this bit to initiate another BIST FIS.	0	RW
08	P0BFI	**Port 0 BIST FIS Initiate:** 0 = Disable. 1 = When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 0, using the parameters specified in this register and BFTD1 and BFTD2. The BIST FIS will only be initiated if a device is present and not in the partial or slumber states. Software must clear this bit before disabling port 0. After a BIST FIS is successfully completed, software must disable and reenable PCS.P0E prior to attempting additional BIST FISes or to return the SATA controller to a normal operational mode. If the BIST FIS fails, as indicated by BFF in this register, software can clear then set this bit to initiate another BIST FIS.	0	RW
07:02	BFP	**BIST FIS Parameters:** These bits form the contents of the upper 6 bits of the BIST FIS Pattern Definition in the BIST FIS transmitted by the SATA controller. This field is not port specific – its contents is used for any BIST FIS initiated on the SATA controller. The specific bit definitions are: Bit Symbol Description 7 T Far End Transmit mode 6 A Align Bypass mode 5 S Bypass Scrambling 4 L Far End Retimed Loopback 3 F Far End Analog Loopback 2 P Primitive bit for use with Transmit mode	00	RW
01:00	Reserved	Reserved	00	

29.1.6.11 Offset E4 - E7h: BFTD1 – BIST FIS Transmit Data 1 Register

Table 922. Offset E4 - E7h: BFTD1 – BIST FIS Transmit Data 1 Register

Device: 31	**Function:** 2
Offset: E4 - E7h	**Size:** 32 bit
Default Value: 00000000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:00	DATA	**BIST FIS Transmit Data 1:** The data programmed into this register will form the contents of the second Dword of any BIST FIS initiated by the SATA controller. This register is not port specific – its contents is used for BIST FIS initiated on any port. Although the second and third Dwords of the BIST FIS are only meaningful when the T bit of the BIST FIS is set to indicate "Far-End Transmit mode", this register's contents are transmitted as the BIST FIS second Dword regardless of whether or not the T bit is indicated in the BFCS register.	0	RW

29.1.6.12 Offset E8 - EBh: BFTD2 – BIST FIS Transmit Data 2 Register

Table 923. Offset E8 - EBh: BFTD2 – BIST FIS Transmit Data 2 Register

Device: 31	**Function:** 2
Offset: E8 - EBh	**Size:** 32 bit
Default Value: 00000000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:00	DATA	**BIST FIS Transmit Data 2:** The data programmed into this register will form the contents of the third Dword of any BIST FIS initiated by the SATA controller. This register is not port specific – its contents is used for BIST FIS initiated on any port. Although the second and third Dwords of the BIST FIS are only meaningful when the "T" bit of the BIST FIS is set to indicate "Far-End Transmit mode", this register's contents are transmitted as the BIST FIS third Dword regardless of whether or not the "T" bit is set in the BFCS register.	0	RW

29.1.6.13 Offset F8h: MID – Manufacturing ID Register

Table 924. Offset F8h: MID – Manufacturing ID Register

Device: 31	**Function:** 2
Offset: F8h	**Size:** 32 bit
Default Value: 00010F80h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:24	Reserved	Reserved	00h	
23:16	SID	**Stepping Identifier:** This field increments for each stepping of the part. This field can be used by software to differentiate steppings when the Revision ID may not change. See Section 16.2.4 for cases in which the Revision ID may not increment. Implementation Note: A single Stepping ID can be implemented that is readable from all functions in the chip because all of them increment in lock-step.	01h - A1	RO
15:08	MID	**Manufacturing Identifier:** 0Fh = Intel	0Fh	RO
07:00	Reserved	Reserved.	80h	

29.2 I/O Registers

All I/O registers are in the core well. These registers use 16 bytes of I/O space in legacy mode and 32 bytes of I/O space when the sub-class code is SATA and PI as AHCI . These I/O registers are allocated via the LBAR register (configuration offset 20h).

When the sub-class code is SATA and PI as AHCI, Index/Data pair mechanism is used for indirect AHCI register accesses. This mechanism allows host software access to all the AHCI registers using indirect I/O addressing without having to use the AHCI BAR (Base Address Register). Only BIOS and OROM must use the Index/Data pair mechanism during boot up. The AHCI device driver must not use the Index/Data pair mechanism to access AHCI registers, otherwise the results are unpredictable. The AHCI device drivers must use the ABAR to access AHCI registers. After using ABAR to access AHCI registers, Index/Data pair mechanism should not be used as this will no longer be functional.

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 925. I/O Registers Summary Table

Start	End	Symbol	Name	Default	Access
00h	00h	PCMD	Primary Command	00h	RO,RW
02h	02h	PSTS	Primary Status	00h	RO, RW, RWC
04h	07h	PDTP	Primary Descriptor Table Pointer	XXh	RO, RW
08h	08h	SCMD	Secondary Command	00h	RO,RW
0Ah	0Ah	SSTS	Secondary Status	00h	RO, RW, RWC
0Ch	0Fh	SDTP	Secondary Descriptor Table Pointer	XXh	RO, RW
10h	13h	INDEX	AHCI Index register	00000000h	RO, RW
14h	17h	DATA	AHCI Data register	See register description	RW

29.2.1 Primary Devices

29.2.1.1 Offset 00h: PCMD – Primary Command Register

Table 926. Offset 00h: PCMD – Primary Command Register (Sheet 1 of 2)

I/O Address: 00h ***Size:*** 8 bit

Default Value: *00h*

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	0	

Table 926. Offset 00h: PCMD – Primary Command Register (Sheet 2 of 2)

I/O Address: 00h		*Size:* 8 bit		
Default Value: 00h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
03	RWC	**Read/Write Control:** Sets the direction of the bus master transfer: 0 = Memory to device. 1 = Device to memory. This bit must not be changed when the bus master function is active.	0	RW
02:01	Reserved	Reserved	0	
00	START	**Start/Stop Bus Master:** 0 = All state information is lost when this bit is written to '0'; Master mode operation cannot be stopped and then resumed. If this bit is reset while bus master operation is still active and the device has not yet finished its data transfer, the bus master command is said to be aborted. 1 = Setting this bit enables bus master operation of the controller. Bus master operation does not actually start unless the Bus Master Enable bit in PCI configuration space is also set. Clearing it halts bus master operation.	0	RW

29.2.1.2 Offset 02h: PSTS – Primary Status Register

Table 927. Offset 02h: PSTS – Primary Status Register (Sheet 1 of 2)

I/O Address: 02h		*Size:* 8 bit		
Default Value: 00h				
Lockable:		*Power Well:*		
Bits	**Name**	**Description**	**Reset Value**	**Access**
07	PRDIS	**PRD Interrupt Status:** 0 = Software clears this bit by writing a 1 to it. 1 = This bit is set when the host controller executes a PRD that has its PRD_INT bit set.	0	RWC
06	D1DC	**Device 1 DMA Capable:** 0 = Disable 1 = A scratchpad bit set software to indicate that device 1 of this channel is capable of DMA transfers. This bit has no effect on the hardware.	0	RW
05	D0DC	**Device 0 DMA Capable:** 0 = Disable 1 = A scratchpad bit set software to indicate that device 0 of this channel is capable of DMA transfers. This bit has no effect on the hardware.	0	RW
04:03	Reserved	Reserved	0	

Table 927. Offset 02h: PSTS – Primary Status Register (Sheet 2 of 2)

I/O Address: 02h			***Size:*** 8 bit	
Default Value: *00h*				
Lockable:			***Power Well:***	
Bits	**Name**	**Description**	**Reset Value**	**Access**
02	I	**Interrupt:** 0 = Software clears this bit by writing a 1 to it. 1 = Set when a device FIS is received with the '1' bit set, provided that software has not disabled interrupts via the nIEN bit of the Device Control Register (see chapter 5 of the *Serial ATA Specification, Rev. 1.0a*).	0	RWC
01	ERR	**Error:** 0 = Software clears this bit by writing a 1 to it. 1 = This bit is set when the controller encounters an error during the transfer and must stop the transfer. See Section 29.5.4 for the list of errors that set this bit.	0	RWC
00	ACT	**Active:** 0 = Cleared by the host when the last transfer for a region is performed, where EOT for that region is set in the region descriptor, and when the START bit is cleared in the Command register and the controller has returned to an idle condition. 1 = Set by the host when the START bit is written to the Command register.	0	RO

29.2.1.3 Offset 04 - 07h: PDTP – Primary Descriptor Table Pointer Register

Table 928. Offset 04 - 07h: PDTP – Primary Descriptor Table Pointer Register

I/O Address: 04 - 07h			***Size:*** 32 bit	
Default Value: XXh				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:02	DBA	**Descriptor Base Address:** Corresponds to A[31:2]. This table must not cross a 64 Kbyte boundary in memory. When read, the current value of the pointer is returned.	XXh	RW
01:00	Reserved	Reserved	00	

29.2.2 Secondary Devices

29.2.2.1 Offset 08h: SCMD – Secondary Command Register

See the description for the PCMD register Table 29.2.1.1 on page 917.

29.2.2.2 Offset 0Ah: SSTS – Secondary Status Register

See the description for the PSTS register Table 29.2.1.2 on page 918.

29.2.2.3 Offset 0C - 0Fh: SDTP – Secondary Descriptor Table Pointer Register

See the description for the PDTP register Table 29.2.1.3 on page 919.

29.2.3 Indirect AHCI Addressing Index/Data registers

29.2.3.1 Offset 10 - 13h: INDEX – Index Register

Table 929. Offset 10 - 13h: INDEX – Index Register

I/O Address: 10 - 13h		***Size:*** 32 bit		
Default Value: *00000000h*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:10	Reserved	Reserved	000000h	
09:02	INDEX	**Index:** This index register is used to select the Dword offset of the Memory Mapped AHCI register to be accessed. A Dword, Word or Byte access is specified by the active byte enables of the I/O access to the Data register.	00h	RW
01:00	Reserved	Reserved	0h	

29.2.3.2 Offset 14 - 17h: DATA – Data Register

Table 930. Offset 14 - 17h: DATA – Data Register

Offset: 14 - 17h		***Size:*** 32 bit		
Default Value: *See register description*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:00	DATA	**Data:** The data register is a "window" through which data is read or written to the memory mapped register pointed to by the index register. A physical register is not actually implemented as the data is actually stored in the memory mapped registers. Since this is not a physical register, the "default" value is the same as the default value of the register pointed to by index.	Same as the register pointed to by index	RW

29.3 Memory Registers

The memory mapped registers within the SATA controller exist in non-cacheable memory space. Additionally, locked accesses are not supported. If software attempts to perform locked transactions to the registers, indeterminate results may occur. Register accesses shall have a maximum size of 64 bits. 64 bit accesses must not cross an 8 byte alignment boundary.

The registers are broken into two sections - global control registers and port control registers. All registers that start below address 100h are global and meant to apply to the entire Host Bus Adaptor (HBA). The port control registers are the same for all ports, and there are as many registers banks as there are ports.

All registers not defined and all reserved bits within registers return '0' when read.

Table 931. Memory Register Summary Table

Start	End	Description
00h	1F	Generic Host Control Register
20h	9F	Reserved
A0h	FF	Vendor Specific Registers

Table 931. Memory Register Summary Table

100h	17F	Port 0 port control Registers
180h	1FF	Port 1 port control Registers
200h	27F	Port 2 port control Registers.
280h	2FF	Port 3 port control Registers.
300h	37F	Port 4 port control Registers.
380h	3FF	Port 5 port control Registers.

29.3.1 Generic Host Controller

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 932. Generic Host Controller Register Summary Table

Start	End	Symbol	Description	Default	Access
00h	03h	CAP	Host Capabilities Register	C6127F05h	RO, RWOnce
04h	07h	GHC	Global Host Control Register	00000000h	RO, RW
08h	0Bh	IS	Interrupt Status Register	00000000h	RO, RWC
0Ch	0Fh	PI	Ports Implemented Register	00000000h	RO, RWOnce
10h	13h	VS	Version Register	00010100h	RO

29.3.1.1 Offset 00 - 03h: CAP – HBA Capabilities Register

This register indicates basic capabilities of the HBA to driver software. The RWO bits in this register are only cleared upon PLTRST#.

Table 933. Offset 00 - 03h: CAP – HBA Capabilities Register (Sheet 1 of 2)

I/O Address: *00 - 03h* ***Size:*** 32 bit

Default Value: C6127F05h

Bits	Name	Description	Reset Value	Access
31	S64A	**Supports 64-bit Addressing:** 0 = 64-bit Addressing not supported. 1 = Indicates the SATA controller can access 64-bit data structures. The 32-bit upper bits of the port DMA Descriptor, the PRD Base, and each PRD entry are read/write.	1	RO
30	SCQA	**Supports Native Command Queue Acceleration:** 0 = Native Command Queue Acceleration not supported. 1 = Indicates the SATA controller supports Serial-ATA native command queuing. The HBA will handle DMA Setup FISes natively and will handle the auto-activate optimization through the FIS.	1	RWO
29	Reserved	**Reserved**	0	
28	SIS	The SATA controller does not support interlock switch.	0	RWO

Table 933. Offset 00 - 03h: CAP – HBA Capabilities Register (Sheet 2 of 2)

I/O Address: *00 - 03h* ***Size:*** 32 bit

Default Value: C6127F05h

Bits	Name	Description	Reset Value	Access
27	SSS	**Supports Staggered Spin-up:** Indicates whether the SATA controller supports staggered spin-up on its ports for use in balancing power spikes. This value is loaded by platform BIOS prior to OS initialization. 0 = Staggered spin-up not supported. 1 = Staggered spin-up supported.	0	RWO
26	SALP	**Supports Aggressive Link Power Management:** 0 = Aggressive Link Power Management not supported. 1 = Indicates the SATA controller supports auto-generating link requests to the partial or slumber states when there are no commands to process.	1	RWO
25	SAL	**Supports Activity LED:** 0 = Activity LED not supported. 1 = Indicates the SATA controller supports a single output pin (SATALED#) which indicates activity.	1	RO
24	Reserved	**Reserved**	0	
23:20	ISS	**Interface Speed Support:** Indicates the maximum speed the SATA controller can support on its ports 1h = 1.5 Gbps	1h	RO
19	SNZO	**Supports Non-Zero DMA Offsets:** Reserved as per the *Advanced Host Controller Interface (AHCI) Specification for Serial ATA, Rev. 1.0*.	0	
18	Reserved	**Reserved**	0	
17	PMS	**Supports Port Multiplier:** The SATA controller does not support command-based switched Port Multipliers. BIOS must set this bit to 0 since Port Multipliers are not supported.	1	RWO
16	PMFS	**Supports Port Multiplier FIS Based Switching:** The SATA controller does not support FIS based switching.	0	RO
15	PMD	**PIO Multiple DRQ Block:** The SATA controller does not support PIO Multiple DRQ Command Block.	0	R0
14	SSC	**Slumber State Capable:** 0 = Slumber state not supported. 1 = The SATA controller supports the slumber state.	1	RWO
13	PSC	**Partial State Capable:** 0 = Partial state not supported. 1 = The SATA controller supports the partial state.	1	RWO
12:08	NCS	**Number of Command Slots:** Support for 32 slots.	1Fh	RO
07:05	Reserved	Reserved	0	
04:00	NDS	**Number of Devices:** Indicates support for six devices.	5h	RO

29.3.1.2 Offset 04 - 07h: GHC – Global HBA Control Register

This register controls various global actions of the HBA.

Table 934. Offset 04 - 07h: GHC – Global HBA Control Register

I/O Address: 04 - 07h ***Size:*** 32 bit

Default Value: *00000000h*

Bits	Name	Description	Reset Value	Access
31	AE	**AHCI Enable:** When set, software shall only talk to the HBA using AHCI. The HBA will not have to allow command processing via both AHCI and legacy mechanisms. When cleared, software will only communicate with the HBA using legacy mechanisms. Software shall set this bit to '1' before accessing other AHCI registers. 0 = Disable 1 = Indicates that an AHCI driver is loaded and communication to the HBA shall be via AHCI mechanisms. This can be used by an HBA that supports both legacy mechanisms (such as SFF-8038i) and AHCI to know when the HBA is running under an AHCI driver.	0	RW
30:02	Reserved	Reserved	0	
01	IE	**Interrupt Enable:** This global bit enables interrupts from the HBA. 0 = All interrupt sources from all ports are enabled. 1 = Interrupts are not allowed.	0	RW
00	HR	**HBA Reset:** 0 = When the HBA has performed the reset action, it will reset this bit to '0'. A software write of '0' will have no effect. For a description on which bits are reset when this bit is set, see the *AHCI Specification*, Section 12.3.3. 1 = When set by software, this bit causes an internal reset of the HBA. All state machines that relate to data transfers and queuing will return to an idle condition, and all ports are reinitialized via COMRESET.	0	RW

29.3.1.3 Offset 08 - 0Bh: IS – Interrupt Status Register

This register indicates which of the ports within the controller have an interrupt pending and require service.

Table 935. Offset 08 - 0Bh: IS – Interrupt Status Register (Sheet 1 of 2)

I/O Address: 08 - 0Bh ***Size:*** 32 bit

Default Value: *00000000h*

Bits	Name	Description	Reset Value	Access
31:06	Reserved	Reserved	0	
05	IPS5	**Interrupt Pending Status Port 5:** 0 = No interrupt pending. 1 = Port 5 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.	0	RWC
04	IPS4	**Interrupt Pending Status Port 4:** 0 = No interrupt pending. 1 = Port 4 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.	0	RWC
03	IPS3	**Interrupt Pending Status Port 3:** 0 = No interrupt pending. 1 = Port 3 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.	0	RWC

Table 935. Offset 08 - 0Bh: IS – Interrupt Status Register (Sheet 2 of 2)

I/O Address: 08 - 0Bh ***Size:*** 32 bit

Default Value: *00000000h*

Bits	Name	Description	Reset Value	Access
02	IPS2	**Interrupt Pending Status Port 2:** 0 = No interrupt pending. 1 = Port 2 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.	0	RWC
01	IPS1	**Interrupt Pending Status Port 1:** 0 = No interrupt pending. 1 = Port 1has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.	0	RWC
00	IPS0	**Interrupt Pending Status Port 0:** 0 = No interrupt pending. 1 = Port 0 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.	0	RWC

29.3.1.4 Offset 0C - 0Fh: PI – Ports Implemented Register

This register indicates which ports are exposed to the HBA. It is loaded by platform BIOS. It indicates which ports that the device supports are available for software to use. Any port may not be implemented.

Table 936. Offset 0C - 0Fh: PI – Ports Implemented Register

I/O Address: 0C - 0Fh ***Size:*** 32 bit

Default Value: *00000000h*

Bits	Name	Description	Reset Value	Access
31:06	Reserved	Reserved	0	
05	PI5	**Port 5 Implemented:** 0 = The port is not supported. 1 = The port is supported.	0	RWO
04	PI4	**Port 4 Implemented:** 0 = The port is not supported. 1 = The port is supported.	0	RWO
03	PI3	**Port 3 Implemented:** 0 = The port is not supported. 1 = The port is supported.	0	RWO
02	PI2	**Port 2 Implemented:** 0 = The port is not supported. 1 = The port is supported.	0	RWO
01	PI1	0 = **Port 1 Implemented:** The port is not supported. 1 = The port is supported.	0	RWO
00	PI0	**Port 0 Implemented:** 0 = The port is not supported. 1 = The port is supported.	0	RWO

29.3.1.5 Offset 10 - 13h: VS – AHCI Version Register

This register indicates the major and minor version of the *AHCI Specification*. It is BCD encoded. The upper two bytes represent the major version number, and the lower two bytes represent the minor version number. Example: Version 3.12 would be represented as 00030102h. The current version of the specification is 1.1 (00010100h).

Table 937. Offset 10 - 13h: VS – AHCI Version Register

I/O Address: 10 - 13h ***Size:*** 32 bit

Default Value: 00010100h

Bits	Name	Description	Reset Value	Access
31:16	MJR	**Major Version Number:** Indicates the major version is “1”	0001h	RO
15:00	MNR	**Minor Version Number:** Indicates the minor version is “10”.	0100h	RO

29.3.2 Vendor Specific Registers

29.3.3 Port DMA Registers

Table 938. Port DMA Registers for Ports[1:0]

Start	End	Symbol	Description	Default	Access
100h	100h	P0CLB	Port 0 Command List Base Address	See register description	RO, RW
104h	104h	P0CLBU	Port 0 Command List Base Address Upper 32-Bits	XXXXXXXXh	RW
108h	108h	P0FB	Port 0 FIS Base Address	See register description	RO, RW
10Ch	10Ch	P0FBU	Port 0 FIS Base Address Upper 32-Bits	XXXXXXXXh	RW
110h	110h	P0IS	Port 0 Interrupt Status	00000000h	RO, RWC
114h	114h	P0IE	Port 0 Interrupt Enable	00000000h	RO, RW
118h	118h	P0CMD	Port 0 Command	See register description	RO, RW
180h	180h	P1CLB	Port 1 Command List Base Address	See register description	RO, RW
184h	184h	P1CLBU	Port 1 Command List Base Address Upper 32-Bits	XXXXXXXXh	RW
188h	188h	P1FB	Port 1 FIS Base Address	See register description	RO, RW
18Ch	18Ch	P1FBU	Port 1 FIS Base Address Upper 32-Bits	XXXXXXXXh	RW
190h	190h	P1IS	Port 1 Interrupt Status	00000000h	RO, RWC
194h	194h	P1IE	Port 1 Interrupt Enable	00000000h	RO, RW
198h	198h	P1CMD	Port 1 Command	See register description	RO, RW

Table 939. Port DMA Registers for Ports[3:2]

Start	End	Symbol	Description	Default	Access
200h	200h	P2CLB	Port 2 Command List Base Address	See register description	RO, RW
204h	204h	P2CLBU	Port 2 Command List Base Address Upper 32-Bits	XXXXXXXXh	RW
208h	208h	P2FB	Port 2 FIS Base Address	See register description	RO, RW
20Ch	20Ch	P2FBU	Port 2 FIS Base Address Upper 32-Bits	XXXXXXXXh	RW
210h	210h	P2IS	Port 2 Interrupt Status	00000000h	RO, RWC
214h	214h	P2IE	Port 2 Interrupt Enable	00000000h	RO, RW
218h	218h	P2CMD	Port 2 Command	See register description	RO, RW
280h	280h	P3CLB	Port 3 Command List Base Address	See register description	RO, RW
284h	284h	P3CLBU	Port 3 Command List Base Address Upper 32-Bits	XXXXXXXXh	RW
288h	288h	P3FB	Port 3 FIS Base Address	See register description	RO, RW
28Ch	28Ch	P3FBU	Port 3 FIS Base Address Upper 32-Bits	XXXXXXXXh	RW
290h	290h	P3IS	Port 3 Interrupt Status	00000000h	RO, RWC
294h	294h	P3IE	Port 3 Interrupt Enable	00000000h	RO, RW
298h	298C	P3CMD	Port 3 Command	See register description	RO, RW

Table 940. Port DMA Registers for Ports[5:4]

Start	End	Symbol	Description	Default	Access
300h	300h	P4CLB	Port 4 Command List Base Address	See register description	RO, RW
304h	304h	P4CLBU	Port 4 Command List Base Address Upper 32-Bits	XXXXXXXXh	RW
308h	308h	P4FB	Port 4 FIS Base Address	See register description	RO, RW
30Ch	30Ch	P4FBU	Port 4 FIS Base Address Upper 32-Bits	XXXXXXXXh	RW
310h	310h	P4IS	Port 4 Interrupt Status	00000000h	RO, RWC
314h	314h	P4IE	Port 4 Interrupt Enable	00000000h	RO, RW
318h	318h	P4CMD	Port 4 Command	See register description	RO, RW
380h	380h	P5CLB	Port 5 Command List Base Address	See register description	RO, RW
384h	384h	P5CLBU	Port 5 Command List Base Address Upper 32-Bits	XXXXXXXXh	RW
388h	388h	P5FB	Port 5 FIS Base Address	See register description	RO, RW
38Ch	38Ch	P5FBU	Port 5 FIS Base Address Upper 32-Bits	XXXXXXXXh	RW

Table 940. Port DMA Registers for Ports[5:4]

390h	390h	P5IS	Port 5 Interrupt Status	00000000h	RO, RWC
394h	394h	P5IE	Port 5 Interrupt Enable	00000000h	RO, RW
398h	398h	P5CMD	Port 5 Command	See register description	RO, RW

29.3.3.1 Offset 100h, 180h, 200h, 280h, 300h, 380h: PxCLB – Port [0-5] Command List Base Address Register

Table 941. Offset 100h, 180h, 200h, 280h, 300h, 380h: PxCLB – Port [0-5] Command List Base Address Register

I/O Address: 100h, 180h, 200h, 280h, 300h, 380h ***Size:*** 32 bit

Default Value: *See description below*

Bits	Name	Description	Reset Value	Access
31:10	CLB	**Command List Base Address:** Indicates the 32-bit base for the command list for this port. This base is used when fetching commands to execute. This address must be 1-KB aligned as indicated by bits 31:10 being read/write. ***Note:*** These bits are not reset on an HBA reset.	XXh.	RW
09:00	Reserved	Reserved	000	

29.3.3.2 Offset 104h, 184h, 204h, 284h, 304h, 384h: PxCLBU – Port [0-5] Command List Base Address Upper 32-bits Register

Table 942. Offset 104h, 184h, 204h, 284h, 304h, 384h: PxCLBU – Port [0-5] Command List Base Address Upper 32-bits Register

I/O Address: 104h, 184h, 204h, 284h, 304h, 384h ***Size:*** 32 bit

Default Value: *XXXXXXXXh*

Bits	Name	Description	Reset Value	Access
31:00	CLBU	**Command List Base Address Upper:** Indicates the upper 32-bits for the command list base address for this port. This base is used when fetching commands to execute. ***Note:*** These bits are not reset on an HBA reset.	XXh	RW

29.3.3.3 Offset 108h, 188h, 208h, 288h, 308h, 388h: PxFB – Port [0-5] FIS Base Address Register

Table 943. Offset 108h, 188h, 208h, 288h, 308h, 388h: PxFB – Port [0-5] FIS Base Address Register

I/O Address: 108, 188h, 208h, 288h, 308h, 388h ***Size:*** 32 bit

Default Value: *See description below*

Bits	Name	Description	Reset Value	Access
31:08	FB	**FIS Base Address:** Indicates the 32-bit base for received FISes. This address must be 256-byte aligned as indicated by bits 31:08 being read/write. ***Note:*** These bits are not reset on an HBA reset.	XXh	RW
07:00	Reserved	Reserved	000	

29.3.3.4 Offset 10Ch, 18Ch, 20Ch, 28Ch, 30Ch, 38Ch: PxFBU – Port [0-5] FIS Base Address Upper 32-bits Register

Table 944. Offset 10Ch, 18Ch, 20Ch, 28Ch, 30Ch, 38Ch: PxFBU – Port [0-5] FIS Base Address Upper 32-bits Register

I/O Address: 10Ch, 18Ch, 20Ch, 28Ch, 30Ch, 38Ch ***Size:*** 32 bit

Default Value: *XXXXXXXXh*

Bits	Name	Description	Reset Value	Access
31:00	FBU	**FIS Base Address Upper:** Indicates the upper 32-bits for the received FIS base for this port. ***Note:*** These bits are not reset on an HBA reset.	XXh	RW

29.3.3.5 Offset 110h, 190h, 210h, 290h, 310h, 390h: PxIS – Port [0-5] Interrupt Status Register

Table 945. Offset 110h, 190h, 210h, 290h, 310h, 390h: PxIS – Port [0-5] Interrupt Status Register (Sheet 1 of 3)

I/O Address: 110, 190h, 210h, 290h, 310h, 390h ***Size:*** 32 bit

Default Value: *00000000h*

Bits	Name	Description	Reset Value	Access
31	CPDS	**Cold Port Detect Status:** The SATA controller does not support cold presence detect.	0	RO
30	TFES	**Task File Error Status:** 0 = Set to 0. 1 = This bit is set whenever the status register is updated by the device and the error bit (bit 0) is set.	0	RWC
29	HBFS	**Host Bus Fatal Error Status:** 0 = Set to 0. 1 = Indicates that the HBA encountered a host bus error that it cannot recover from, such as a bad software pointer. In PCI, such an indication would be a target or master abort.	0	RWC

Table 945. Offset 110h, 190h, 210h, 290h, 310h, 390h: PxIS – Port [0-5] Interrupt Status Register (Sheet 2 of 3)

I/O Address: 110, 190h, 210h, 290h, 310h, 390h ***Size:*** 32 bit

Default Value: *00000000h*

Bits	Name	Description	Reset Value	Access
28	HBDS	***Host Bus Data Error Status:*** 0 = Set to 0. 1 = Indicates that the HBA encountered a data error (uncorrectable ECC/parity) when reading from or writing to system memory.	0	RWC
27	IFS	**Interface Fatal Error Status:** 0 = Set to 0. 1 = Indicates that the HBA encountered an error on the SATA interface which caused the transfer to stop.	0	RWC
26	INFS	**Interface Non-fatal Error Status:** 0 = Set to 0. 1 = Indicates that the Host Bus Adaptor encountered an error on the SATA interface but was able to continue operation.	0	RWC
25	Reserved	**Reserved**	0	
24	OFS	**Overflow Status:** 0 = Set to 0. 1 = Indicates that the HBA received more bytes from a device than was specified in the PRD table for the command.	0	RWC
23	IPMS	**Incorrect Port Multiplier Status:** 0 = Set to 0. 1 = Indicates that the HBA received a FIS from a device whose port multiplier field did not match what was expected.	0	RWC
22	PRCS	**PhyRdy Change Status:** 0 = Set to 0. Indicates the internal PhyRdy signal Changed state. This bit reflects the state of PxSERR.DIAG.N. Unlike most of the other bits in the register, this bit is RO and is only cleared when PxSERR.DIAG.N is cleared. The internal PhyRdy signal also transitions when the port interface enters PARTIAL or SLUMBER power management states. PARTIAL or SLUMBER must be disabled when Surprise Removal Notification is desired; otherwise, the power management state transitions will appear as false insertion and removal events.	0	RO
21:08	Reserved	Reserved	0	
07	Reserved	**Reserved**	0	
06	PCS	**Port Connect Change Status:** 0 = No change in Current Connect Status. 1 = Change in Current Connect Status. This bit reflects the state of PxSERR.DIAG.X. Unlike other bits in this register, this bit is only cleared when PxSERR.DIAG.X is cleared.	0	RO
05	DPS	**Descriptor Processed:** 0 = No change in Descriptor process status. 1 = A PRD with the '1' bit set has transferred all of its data.	0	RWC
04	UFS	**Unknown FIS Interrupt:** When set to '1' 0 = Indicates no unknown FIS interrupt. 1 = Indicates that an unknown FIS was received and has been copied into system memory. This bit is cleared to '0' by software clearing the PxSERR.DIAG.F bit to '0'. Note that this bit does not directly reflect the PxSERR.DIAG.F bit. PxSERR.DIAG.F is set immediately when an unknown FIS is detected, whereas this bit is set when the FIS is posted to memory. Software must wait to act on an unknown FIS until this bit is set to '1' or the two bits may become out of sync.	0	RO

Table 945. Offset 110h, 190h, 210h, 290h, 310h, 390h: PxIS – Port [0-5] Interrupt Status Register (Sheet 3 of 3)

I/O Address: 110, 190h, 210h, 290h, 310h, 390h		***Size:*** 32 bit		
Default Value: *00000000h*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
03	SDBS	**Set Device Bits Interrupt:**. 0 = Indicates no set device bits interrupt. 1 = A Set Device Bits FIS has been received with the '1' bit set and has been copied into system memory.	0	RWC
02	DSS	**DMA Setup FIS Interrupt:** 0 = Indicates no DMA Setup FIS interrupt. 1 = A DMA Setup FIS has been received with the '1' bit set and has been copied into system memory.	0	RWC
01	PSS	**PIO Setup FIS Interrupt:** 0 = Indicates no PIO Setup FIS interrupt. 1 = A PIO Setup FIS has been received with the '1' bit set, it has been copied into system memory, and the data related to that FIS has been transferred. This bit is set even if the data transfer resulted in an error.	0	RWC
00	DHRS	**Device to Host Register FIS Interrupt:** 0 = Indicates no Device to Host Register FIS interrupt. 1 = A D2H register FIS has been received with the '1' bit set and has been copied into system memory.	0	RWC

29.3.3.6 Offset 114h, 194h, 214h, 294h, 314h, 394h: PxIE – Port [0-5] Interrupt Enable Register

This register enables and disables the reporting of the corresponding interrupt to system software. When a bit is set, logic 1, and the corresponding interrupt condition is active, then an interrupt is generated. Interrupt sources that are disabled, logic 0, are still reflected in the status registers. This register is symmetrical with the Port 0 Status register.

Table 946. Offset 114h, 194h, 214h, 294h, 314h, 394h: PxIE – Port [0-5] Interrupt Enable Register (Sheet 1 of 2)

I/O Address: 114h, 194h, 214h, 294h, 314h, 394h		***Size:*** 32 bit		
Default Value: *00000000h*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31	CPDE	**Cold Presence Detect Enable:** The SATA controller does not support cold presence detect.	0	RO
30	TFEE	**Task File Error Enable:** 0 = Disable. 1 = When set, GHC.IE is set, and P0S.TFES is set, the HBA shall generate an interrupt.	0	RW
29	HBFE	**Host Bus Fatal Error Enable:** 0 = Disable. 1 = When set, GHC.IE is set, and P0IS.HBFS is set, the HBA shall generate an interrupt.	0	RW
28	HBDE	**Host Bus Data Error Enable:** 0 = Disable. 1 = When set, GHC.IE is set, and P0IS.HBDS is set, the HBA shall generate an interrupt.	0	RW

Table 946. Offset 114h, 194h, 214h, 294h, 314h, 394h: PxIE – Port [0-5] Interrupt Enable Register (Sheet 2 of 2)

I/O Address: 114h, 194h, 214h, 294h, 314h, 394h **Size:** 32 bit

Default Value: *00000000h*

Bits	Name	Description	Reset Value	Access
27	IFE	**Interface Fatal Error Enable:** 0 = Disable. 1 = When set, GHC.IE is set, and P0IS.IFS is set, the HBA shall generate an interrupt.	0	RW
26	INFE	**Interface Non-fatal Error Enable:** 0 = Disable. 1 = When set, GHC.IE is set, and P0IS.INFS is set, the HBA shall generate an interrupt.	0	RW
25	Reserved	Reserved	0	
24	OFE	**Overflow Enable:** 0 = Disable. 1 = When set, and GHC.IE and P0IS.OFS are set, the HBA shall generate an interrupt.	0	RW
23	IPME	**Incorrect Port Multiplier Enable:** 0 = Disable. 1 = When set, and GHC.IE and P0IS.IPMS are set, the HBA shall generate an interrupt.	0	RW
22	PRCE	**PhyRdy Change Interrupt Enable:** 0 = Disable. 1 = When set, and GHC.IE is set, and PxIS.PRCS is set, the HBA shall generate an interrupt.	0	RW
21:07	Reserved	Reserved	0	
06	PCE	**Port Change Interrupt Enable:** 0 = Disable. 1 = When set, GHC.IE is set, and P0IS.PCS is set, the HBA shall generate an interrupt.	0	RW
05	DPE	**Descriptor Processed Interrupt Enable:** 0 = Disable. 1 = When set, GHC.IE is set, and P0IS.DPS is set, the HBA shall generate an interrupt.	0	RW
04	UFE	**Unknown FIS Interrupt Enable:** 0 = Disable. 1 = When set, GHC.IE is set, and PxIS.UFS is set to '1', the HBA shall generate an interrupt.	0	RW
03	SDBE	**Set Device Bits FIS Interrupt Enable:** 0 = Disable. 1 = When set, GHC.IE is set, and P0IS.SDBS is set, the HBA shall generate an interrupt.	0	RW
02	DSE	**DMA Setup FIS Interrupt Enable:** 0 = Disable. 1 = When set, GHC.IE is set, and P0IS.DSS is set, the HBA shall generate an interrupt.	0	RW
01	PSE	**PIO Setup FIS Interrupt Enable:** 0 = Disable. 1 = When set, GHC.IE is set, and P0IS.PSS is set, the HBA shall generate an interrupt.	0	RW
00	DHRE	**Device to Host Register FIS Interrupt Enable:** 0 = Disable. 1 = When set, GHC.IE is set, and P0IS.DHRS is set, the HBA shall generate an interrupt.	0	RW

29.3.3.7 Offset 118h, 198h, 218h, 298h, 318h, 398h: PxCMD – Port [0-5] Command Register

Table 947. Offset 118h, 198h, 218h, 298h, 318h, 398h: PxCMD – Port [0-5] Command Register (Sheet 1 of 3)

I/O Address: 118, 198h, 218h, 298h, 318h, 398h ***Size:*** 32 bit

Default Value: *See register description*

Bits	Name	Description	Reset Value	Access
31:28	ICC	**Interface Communication Control:** This is a four bit field which can be used to control reset and power states of the interface. If the Link layer is currently in the L_IDLE state, writes to this field shall cause the HBA to initiate a transition to the interface power management state requested. If the Link layer is not currently in the L_IDLE state, writes to this field shall have no effect. Value — Definition Fh–7h — Reserved 6h — Slumber: This will cause the HBA to request a transition of the interface to the slumber state. The SATA device may reject the request and the interface will remain in its current state 5h–3h — Reserved 2h — Partial: This will cause the HBA to request a transition of the interface to the partial state. The SATA device may reject the request and the interface will remain in its current state. 1h — Active: This will cause the HBA to request a transition of the interface into the active 0h — No-Op / Idle: When software reads this value, it indicates the HBA is ready to accept a new interface control command, although the transition to the previously selected state may not yet have occurred. When system software writes a non-reserved value other than No-Op (0h), the HBA will perform the action and update this field back to Idle (0h). If software writes to this field to change the state to a state the link is already in (i.e., interface is in the active state and a request is made to go to the active state), the HBA will take no action and return this field to Idle. If the interface is in a low power state and the software wants to transition to a different low power state, software must first bring the link to active and then initiate the transition to the desired low power state.	0h	RW
27	ASP	**Aggressive Slumber/Partial:** 0 = When cleared, and the ALPE bit is set, the HBA will aggressively enter the partial state when it clears the PxCI register and the PxSACT register is cleared. 1 = When set, and the ALPE bit is set, the HBA will aggressively enter the slumber state when it clears the PxCI register and the PxSACT register is cleared.	0	RW
26	ALPE	**Aggressive Link Power Management Enable:** 0 = Disable 1 = When set, the HBA will aggressively enter a lower link power state (partial or slumber) based upon the setting of the ASP bit.	0	RW

Table 947. Offset 118h, 198h, 218h, 298h, 318h, 398h: PxCMD – Port [0-5] Command Register (Sheet 2 of 3)

I/O Address: 118, 198h, 218h, 298h, 318h, 398h ***Size:*** 32 bit

Default Value: *See register description*

Bits	Name	Description	Reset Value	Access
25	DLAE	**Drive LED on ATAPI Enable:** 0 = When cleared, the HBA will only drive the LED pin active for commands if PxCMD.ATAPI is set to '0'. 1 = When set, the HBA will drive the LED pin active for commands regardless of the state of PxCMD.ATAPI.	0	RW
24	ATAPI	**Device is ATAPI:** 0 = When cleared, no ATAPI device is connected. 1 = When set, the connected device is an ATAPI device. This bit is used by the HBA to control whether or not to generate the LED when commands are active.	0	RW
23:21	Reserved	Reserved	0	
20	CPD	**Cold Presence Detection:** The SATA controller does not support cold presence detect.	0	RO
19	Reserved	**Reserved**	0	
18	HPCP	**Hot Plug Capable Port:** The HBA takes no action on the state of this bit - it is for system software only. For example, if this bit is cleared, and a hot plug event occurs, the HBA shall still treat it as a proper hot plug event. Note that this bit is not reset on a HBA reset. 0 = Indicates this port is connected to a device that cannot be hot plugged. 1 = This indicates whether the this port is connected to a device which can be hot plugged. SATA by definition is hot-pluggable, but not all platforms are constructed to allow the device to be removed (it may be screwed into the chassis, for example). This bit can be used by system software to indicate a feature such as “eject device” to the end-user.	0	RWO
17	PMA	**Port Multiplier Attached:** This bit is a read only '0' when CAP.PMS = '0', and read/write when CAP.PMS = '1'. 0 = When cleared, a Port Multiplier is not attached to the HBA for this port. 1 = When set, a Port Multiplier is attached to the HBA for this port.	0	RW/RO
16	Reserved	Reserved	0	
15	CR	**Command List Running:** 0 = Set to 0. 1 = When this bit is set it indicates that the command list DMA engine for the port is running.	0	RO
14	FR	**FIS Receive Running:** 0 = Set to 0. 1 = When this bit is set it indicates that the FIS Receive DMA engine for the port is running.	0	RO
13	Reserved	**Reserved**	0	
12:08	CCS	**Current Command Slot:** Indicates the current command slot the HBA is processing. This field is valid when the PxCMD.ST bit is set, and is constantly updated by the HBA. This field can be updated as soon as the HBA recognizes an active command slot, or at some point soon after when it begins processing the command. When PxCMD.ST transitions from a '1' to a '0', the HBA will reset this field to '0'. After PxCMD.ST transitions from '0' to '1', the highest priority slot to issue from next is command slot 0. After the first command has been issued, the highest priority slot to issue from next is PxCMD.CCS + 1. For example, after the HBA has issued its first command, if PxCMD.CCS = 0h and PxCI is set to 3h, the next command that will be issued is from command slot 1.	0h	RO

Table 947. Offset 118h, 198h, 218h, 298h, 318h, 398h: PxCMD – Port [0-5] Command Register (Sheet 3 of 3)

I/O Address: 118, 198h, 218h, 298h, 318h, 398h			***Size:*** 32 bit	
Default Value: *See register description*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:05	Reserved	Reserved	0	
04	FRE	**FIS Receive Enable:** System software must not set this bit until PxFB (PxFBU) have been programmed with a valid pointer to the FIS receive area, If software wishes to move the base, this bit must first be cleared, and software must wait for the PxCMD.FR bit to be cleared. Software must not clear this bit while PxCMD.ST is set to '1'. 0 = When cleared, received FISes are not accepted by the HBA except for the first D2H register FIS after the initialization sequence. 1 = When set to a logic 1, the HBA may post received FISes into the FIS receive area pointed to by PxFB (and for 64-bit HBAs, PxFBU).	0	RW
03	Reserved	Reserved	0	
02	POD	**Power On Device:** The SATA controller does not support cold presence detect.	1	RO
01	SUD	**Spin-Up Device:** This bit is read/write for HBAs that support staggered spin-up via CAP.SSS. This bit is read-only '1' for HBAs that do not support staggered spin-up. On an edge detect from '0' to '1', the HBA shall start a COMRESET initialization sequence to the device. Clearing this bit causes no action on the interface.	X	RW/ RO
00	ST	**Start:** Whenever this bit is changed from a '0' to a '1', the HBA starts processing the command list at entry '0'.Whenever this bit is changed from a '1' to a '0', the PxCI register is cleared by the HBA upon the HBA putting the controller into an idle state. See Section 10.3.1 of the *AHCI Specification* for restrictions on when PxCMD.ST can be set to '1' and cleared to '0'. 0 = When cleared, logic 0, the HBA may not process the command list. 1 = When set, logic 1, the HBA may process the command list.	0	RW

29.3.4 Port Interface Registers (one set per port)

These registers implement various functions of the interface with an SATA device, including the SATA super set registers of the *Serial ATA Specification*.

The port interface registers for port 0, 1, 2 and 3 are always available.

Table 948. Port Interface Registers for Ports[1:0] (Sheet 1 of 2)

Start	End	Symbol	Description	Default	Access
120h	120h	P0TFD	Port 0 Task File Data	00000000h	RO
124h	124h	P0SIG	Port 0 Signature	FFFFFFFFh	RO
128h	128h	P0SSTS	Port 0 Serial ATA Status	00000000h	RO
12Ch	12Ch	P0SCTL	Port 0 Serial ATA Control	00000000h	RO, RW
130h	130h	P0SERR	Port 0 Serial ATA Error	00000000h	RWC
134h	134h	P0SACT	Port 0 Device Status	00000000h	RWOnce
138h	138h	P0CI	Port 0 Command Issue	00000000h	RWOnce
1A0h	1A0h	P1TFD	Port 1 Task File Data	00000000h	RO
1A4h	1A4h	P1SIG	Port 1 Signature	FFFFFFFFh	RO

Table 948. Port Interface Registers for Ports[1:0] (Sheet 2 of 2)

1A8h	1A8h	P1SSTS	Port 1 Serial ATA Status	00000000h	RO
1ACh	1ACh	P1SCTL	Port 1 Serial ATA Control	00000000h	RO, RW
1B0h	1B0h	P1SERR	Port 1 Serial ATA Error	00000000h	RWC
1B4h	1B4h	P1SACT	Port 1 Device Status	00000000h	RWOnce
1B8h	1B8h	P1CI	Port 1 Command Issue	00000000h	RWOnce

Table 949. Port Interface Registers for Ports[3:2]

Start	End	Symbol	Description	Default	Access
220h	220h	P2TFD	Port 2 Task File Data	00000000h	RO
224h	224h	P2SIG	Port 2 Signature	FFFFFFFFh	RO
228h	228h	P2SSTS	Port 2 Serial ATA Status	00000000h	RO
22Ch	22Ch	P2SCTL	Port 2 Serial ATA Control	00000000h	RO, RW
230h	230h	P2SERR	Port 2 Serial ATA Error	00000000h	RWC
234h	234h	P2SACT	Port 2 Device Status	00000000h	RWOnce
238h	238h	P2CI	Port 2 Command Issue	00000000h	RWOnce
2A0h	2A0h	P3TFD	Port 3 Task File Data	00000000h	RO
2A4h	2A4h	P3SIG	Port 3 Signature	FFFFFFFFh	RO
2A8h	2A8h	P3SSTS	Port 3 Serial ATA Status	00000000h	RO
2ACh	2ACh	P3SCTL	Port 3 Serial ATA Control	00000000h	RO, RW
2B0h	2B0h	P3SERR	Port 3 Serial ATA Error	00000000h	RWC
2B4h	2B4h	P3SACT	Port 3 Device Status	00000000h	RWOnce
2B8h	2B8h	P3CI	Port 3 Command Issue	00000000h	RWOnce

Table 950. Port Interface Registers for Ports[5:4]

Start	End	Symbol	Description	Default	Access
320h	320h	P4TFD	Port 4 Task File Data	00000000h	RO
324h	324h	P4SIG	Port 4 Signature	FFFFFFFFh	RO
328h	328h	P4SSTS	Port 4 Serial ATA Status	00000000h	RO
32Ch	32Ch	P4SCTL	Port 4 Serial ATA Control	00000000h	RO, RW
330h	330h	P4SERR	Port 4 Serial ATA Error	00000000h	RWC
334h	334h	P4SACT	Port 4 Device Status	00000000h	RWOnce
338h	338h	P4CI	Port 4 Command Issue	00000000h	RWOnce
3A0h	3A0h	P5TFD	Port 5 Task File Data	00000000h	RO
3A4h	3A4h	P5SIG	Port 5 Signature	FFFFFFFFh	RO
3A8h	3A8h	P5SSTS	Port 5 Serial ATA Status	00000000h	RO
3ACh	3ACh	P5SCTL	Port 5 Serial ATA Control	00000000h	RO, RW
3B0h	3B0h	P5SERR	Port 5 Serial ATA Error	00000000h	RWC
3B4h	3B4h	P5SACT	Port 5 Device Status	00000000h	RWOnce
3B8h	3B8h	P5CI	Port 5 Command Issue	00000000h	RWOnce

29.3.4.1 Offset 120h, 1A0h, 220h, 2A0h, 320h, 3A0h: PxTFD – Port [0-5] Task File Data Register

This is a 32-bit register that copies specific fields of the task file when FISes are received. The FISes that contain this information are:

- D2H Register FIS
- PIO Setup FIS
- Set Device Bits FIS (BSY and DRQ are not updated with this FIS)

Table 951. Offset 120h, 1A0h, 220h, 2A0h, 320h, 3A0h: PxTFD – Port [0-5] Task File Data Register

I/O Address: 120h, 1A0h, 220h, 2A0h,320h,3A0h ***Size:*** 32 bit

Default Value: *00000000h*

Bits	Name	Description	Reset Value	Access
31:16	Reserved	Reserved	0	
15:08	ERR	**Error:** Contains the latest copy of the task file error register.	00h	RO
07:00	STS	**Status:** Contains the latest copy of the task file status register. These fields affect AHCI: Bit / Field / Definition 07 / BSY / Indicates the interface is busy 06:04 / N/A / Not applicable 03 / DRQ / Indicates a data transfer is requested 02:01 / N/A / Not applicable 00 / ERR / Indicates an error during the transfer Note that the HBA updates all 8 bits of this register from the received FIS, not just the bits noted above.	0	RO

29.3.4.2 Offset 124h, 1A4h, 224h, 2A4h, 324h, 3A4h: PxSIG – Port [0-5] Signature Register

This is a 32-bit register that contains the initial signature of an attached device when the first D2H Register FIS is received from that device. It is updated once after a reset sequence.

Table 952. Offset 124h, 1A4h, 224h, 2A4h, 324h, 3A4h: PxSIG – Port [0-5] Signature Register

I/O Address: 124h, 1A4h, 224h, 2A4h,324h,3A4h	***Size:*** 32 bit
Default Value: FFFFFFFFh	

Bits	Name	Description	Reset Value	Access
31:00	SIG	**Signature:** Contains the signature received from a device on the first D2H register FIS. The bit order is as follows: Bit — Field 31:24 — LBA High Register 23:16 — LBA Mid Register 15:08 — LBA Low Register 07:00 — Sector Count Register	FFFFFFFFh	RO

29.3.4.3 Offset 128h, 1A8h, 228h, 2A8h, 328h, 3A8h: PxSSTS – Port [0-5] Serial ATA Status Register

This is a 32-bit register that conveys the current state of the interface and host. The HBA updates it continuously and asynchronously. When the HBA transmits a COMRESET to the device, this register is updated to its reset values.

Table 953. Offset 128h, 1A8h, 228h, 2A8h, 328h, 3A8h: PxSSTS – Port [0-5] Serial ATA Status Register (Sheet 1 of 2)

I/O Address: Offset 128h, 1A8h, 228h, 2A8h, 328h, 3A8h	***Size:*** 32 bit
Default Value: *0000000Xh*	

Bits	Name	Description	Reset Value	Access
31:12	Reserved	Reserved	0	
11:08	IPM	**Interface Power Management:** Indicates the current interface state: 0h Device not present or communication not established 1h Interface in active state 2h Interface in PARTIAL power management state 6h Interface in SLUMBER power management state All other values reserved. This field reflects the interface power management state for both device and host initiated power management.	0	RO
07:04	SPD	**Current Interface Speed:** Indicates the negotiated interface communication speed. 0h Device not present or communication not established 1h Generation 1 communication rate negotiated All other values reserved.	0	RO

Table 953. Offset 128h, 1A8h, 228h, 2A8h, 328h, 3A8h: PxSSTS – Port [0-5] Serial ATA Status Register (Sheet 2 of 2)

I/O Address: Offset 128h, 1A8h, 228h, 2A8h, 328h, 3A8h		***Size:*** 32 bit		
Default Value: *0000000Xh*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
03:00	DET	**Device Detection:** Indicates the interface device detection and Phy state. 0h No device detected and Phy communication not established 1h Device presence detected but Phy communication not established 3h Device presence detected and Phy communication established 4h Phy in offline mode as a result of the interface being disabled or running in a BIST loopback mode All other values reserved. Note that, while the true reset default value of this register is 0h, the value read from this register depends on drive presence and the point in time within the initialization process when the register is read.	X	RO

29.3.4.4 Offset 12Ch, 1ACh, 22Ch, 2ACh, 32Ch, 3ACh: PxSCTL – Port [0-5] Serial ATA Control Register

This is a 32-bit read/write register by which software controls SATA capabilities. Writes to the SControl register result in an action being taken by the host adapter or interface. Reads from this register return the last value written to it.

Table 954. Offset 12Ch, 1ACh, 22Ch, 2ACh, 32Ch, 3ACh: PxSCTL – Port [0-5] Serial ATA Control Register (Sheet 1 of 2)

I/O Address: 12Ch, 1ACh, 22Ch, 2ACh, 32Ch, 3ACh		***Size:*** 32 bit		
Default Value: 00000000h				
Bits	**Name**	**Description**	**Reset Value**	**Access**
31:20	Reserved	Reserved	0	
19:16	PMP	**Port Multiplier Port:** Contains the 4-bit field to insert into outgoing FISes when using a port multiplier. This field is not used by AHCI.	0h	RW
15:12	SPM	**Select Power Management:** Selects a power management state. This field is not used by AHCI HBAs. AHCI HBAs use the PxCMD.ICC to request power state transitions.	0h	RW

Table 954. Offset 12Ch, 1ACh, 22Ch, 2ACh, 32Ch, 3ACh: PxSCTL – Port [0-5] Serial ATA Control Register (Sheet 2 of 2)

I/O Address: 12Ch, 1ACh, 22Ch, 2ACh, 32Ch, 3ACh ***Size:*** 32 bit

Default Value: 00000000h

Bits	Name	Description	Reset Value	Access
11:08	IPM	**Interface Power Management Transitions Allowed:** Indicates which power states the HBA is allowed to transition to If an interface power management state is not allowed via this register field, the HBA will not initiate that state and the HBA will PMNAKP any request from the device to enter that state. 0h No interface restrictions 1h Transitions to the PARTIAL state disabled 2h Transitions to the SLUMBER state disabled 3h Transitions to both PARTIAL and SLUMBER states disabled All other values reserved.	0h	RW
07:04	SPD	**Speed Allowed:** Indicates the highest allowable speed of the interface. 0h No speed negotiation restrictions 1h Limit speed negotiation to Generation 1 communication rate All other values reserved.	0h	RW
03:00	DET	**Device Detection Initialization:** Controls the HBA's device detection and interface initialization. 0h No device detection or initialization action requested. 1h Perform interface communication initialization sequence to establish communication. This is functionally equivalent to a hard reset and results in the interface being reset and communications reinitialized. While this field is 1h, COMRESET is continuously transmitted on the interface. Software must leave the DET field set to 1h for a minimum of 1 millisecond to ensure that a COMRESET is sent on the interface. 4h Disable the Serial ATA interface and put Phy in offline mode. All other values reserved. This field may only be changed to 1h or 4h when PxCMD.ST is '0'. **Changing this field to 1h or 4h while the HBA is running results in undefined behavior.**	0h	RW

29.3.4.5 Offset 130h, 1B0h, 230h, 2B0h, 330h, 3B0h: PxSERR – Port [0-5] Serial ATA Error Register

Table 955. Offset 130h, 1B0h, 230h, 2B0h, 330h, 3B0h: PxSERR – Port [0-5] Serial ATA Error Register (Sheet 1 of 3)

I/O Address: 130h, 1D0h, 230h, 2D0h, 330h, 3B0h ***Size:*** 32 bit

Default Value: *00000000h*

Bits	Name	Description	Reset Value	Access
31:16	DIAG	**Diagnostics:** Contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes: Bits — *Name* — Description 31:27 — Reserved — Reserved 26 — Exchanged (X) — When set to '1' this bit indicates a COMINIT signal was received. This bit is reflected in the interrupt register PxIS. PCS. 25 — Unrecognized FIS Type (F) — Indicates that one or more FISs were received by the Transport layer with good CRC but had a type field that was not recognized/known. 24 — Transport state transition error (T) — Indicates that an error has occurred in the transition from one state to another within the Transport layer since the last time this bit was cleared. 23 — Link Sequence Error (S) — Indicates that one or more Link state machine error conditions were encountered. The Link Layer state machine defines the conditions under which the link layer detects an erroneous transition. 22 — Handshake Error (H) — Indicates that one or more R_ERR handshake response were received in response to frame transmission. Such errors may be the result of a CRC error detected by the recipient, a disparity or 8b/10b decoding error, or other error condition leading to a negative handshake on a transmitted frame. 21 — CRC Error (C) — Indicates that one or more CRC errors occurred with the Link Layer.	0000h	RWC

Table 955. Offset 130h, 1B0h, 230h, 2B0h, 330h, 3B0h: PxSERR – Port [0-5] Serial ATA Error Register (Sheet 2 of 3)

I/O Address: 130h, 1D0h, 230h, 2D0h, 330h, 3B0h ***Size:*** 32 bit

Default Value: *00000000h*

Bits	Name	Description	Reset Value	Access
31:16 (Cont)	DIAG(Cont)	(continued) Bits — *Name* — Description 20 — Disparity Error (D) — Indicates that incorrect disparity was detected one or more times. This field is not used by AHCI. 19 — 10B to 8B Decode Error (B): — Indicates that one or more 10B to 8B decoding errors occurred. 18 — Comm Wake (W) — Indicates that a Comm Wake signal was detected by the Phy. 17 — Phy Internal Error (I) — Indicates that the Phy detected some internal error. 16 — PhyRdy Change (N) — When set to 1 this bit indicates that the internal PhyRdy signal changed state since the last time this bit was cleared. In the IICH, this bit will be set when PhyRdy changes from a 0 -> 1 or a 1 -> 0. The state of this bit is then reflected in the PxIS.PRCS interrupt status bit and an interrupt will be generated if enabled. Software clears this bit by writing a 1 to it.	00h	RWC

Table 955. Offset 130h, 1B0h, 230h, 2B0h, 330h, 3B0h: PxSERR – Port [0-5] Serial ATA Error Register (Sheet 3 of 3)

I/O Address: 130h, 1D0h, 230h, 2D0h, 330h, 3B0h ***Size:*** 32 bit

Default Value: *00000000h*

Bits	Name	Description	Reset Value	Access
15:00	ERR	**Error:** The ERR field contains error information for use by host software in determining the appropriate response to the error condition. If one or more of bits 11:8 of this register are set, the controller will stop the current transfer. Bits — *Name* — Description 15:12 — Reserved — Reserved 11 — Internal Error (E) — The SATA controller failed due to a master or target abort when attempting to access system memory. 10 — Protocol Error (P) — A violation of the Serial ATA protocol was detected. 9 — Persistent Communication or Data Integrity Error (C): — A communication error that was not recovered occurred that is expected to be persistent. Persistent communications errors may arise from faulty interconnect with the device, from a device that has been removed or has failed, or a number of other causes. 8 — Transient Data Integrity Error (T) — A data integrity error occurred that was not recovered by the interface. 7:2 — Reserved — Reserved 1 — Recovered Communications Error (M) — Communications between the device and host were temporarily lost but were reestablished. This can arise from a device temporarily being removed, from a temporary loss of Phy synchronization, or from other causes, and may be derived from the PhyNRdy signal between the Phy and Link layers. 0 — Recovered Data Integrity Error (I) — A data integrity error occurred that was recovered by the interface through a retry operation or other recovery action.	0000h	RWC

29.3.4.6 Offset 134h, 1B4h, 234h, 2B4h, 334h, 3B4h: PxSACT – Port [0-5] Serial ATA Active Register

Table 956. Offset 134h, 1B4h, 234h, 2B4h, 334h, 3B4h: PxSACT – Port [0-5] Serial ATA Active Register

I/O Address: 134h, 1D4h, 234h, 2D4h, 334h, 3B4h	***Size:*** 32 bit
Default Value: *00000000h*	

Bits	Name	Description	Reset Value	Access
31:00	DS	**Device Status:** System software sets this bit for random queuing operations prior to setting the PxCI.CI bit in the same command slot entry. This field is cleared via the Set Device Bits FIS. This field is also cleared when PxCMD.ST is cleared by software. Note that this field is not cleared by COMRESET or SRST.	0	R/W1

29.3.4.7 Offset 138h, 1B8h, 238h, 2B8h, 338h, 3B8h: PxCI – Port [0-5] Command Issue Register

Table 957. Offset 138h, 1B8h, 238h, 2B8h, 338h, 3B8h: PxCI – Port [0-5] Command Issue Register

I/O Address: Offset 138h, 1D4h, 234h, 2D4h, 338h, 3B8h	***Size:*** *32 bit*
Default Value: *00000000h*	

Bits	Name	Description	Reset Value	Access
31:00	CI	**Commands Issued:** This field is set by software to indicate to the HBA that a command has been built in system memory for a command slot and may be sent to the device. When the HBA receives a FIS which clears the BSY, ERR, and DRQ bits for the command, it clears the corresponding bit in this register for that command slot. This field is also cleared when PxCMD.ST is written from a '1' to a '0' by software.	0	RWO

29.4 Overview

The SATA host controller contains two modes of operation – a legacy mode using I/O space, and an AHCI mode using memory space. The memory space bit GHC.AE (see Table 934), set by software, indicates to hardware that AHCI is being used. Software must not implement code that mixes the use of legacy mode and AHCI mode.

29.5 Legacy Operation

The Intel® 3100 Chipset supports standard ATA emulation. In this mode of operation, software is performing I/O operations to the controller and SATA devices. The SATA controller is using the shadow registers as described in the *Serial ATA Specification* and performing master/slave operation on the ports. Software must program the DEV bit in the task file as its first operation before programming the rest of the transfer or setting the bus master registers.

29.5.1 Transfer Examples

29.5.1.1 Register FIS Only

If software sends a command to the SATA device and does not want any ATAPI or data transfer to occur, it performs PIO operations, setting up the required fields for the command, and either writes to the command register (1F7h for primary, 177h for secondary) or control register (3F6h for primary, 376h for secondary).

Hardware will send the register FIS with the appropriate field set for command or control block, and upon reception of the device-to-host register FIS indicating command completion, will update the shadow block.

29.5.1.2 Non-Queued DMA Data Transfers

The following sections describe a data transfer to and from a device using DMA as the command type. These transfers are all non-queued.

29.5.1.2.1 ATA – Data from Memory to Device

Part I - Software Actions – Command Start

1. Software sets up a PRD table in memory, with one or more entries to accomplish the data transfer of the full command. Software places the address for this table PRD Base register.
2. Software writes to the task file to set up the command, with the final write being to the command register (1F7h, 177h).
3. Software sets the “start” bit in the SATA host controller. This step can be swapped with the previous step, since no action is taken until the data transfer is to begin.

Part II - Hardware Actions – Command Start

Upon seeing the command register written, hardware sends the register FIS to the device and awaits reception of a PIO setup FIS, device-to-host register FIS, DMA Activate FIS, or DATA FIS.

Part III - Hardware Actions – Data Transfer

1. Since the direction of the transfer is from the host to the device (and the command was a DMA command), hardware will receive a DMA Activate FIS.
2. Upon reception, if the ‘start’ bit has been set, hardware begins fetching data from the locations specified in the PRD table. (Hardware could, and for performance reasons must, have started fetching data prior to seeing the device-to-host register FIS described in Part II).
3. Hardware formulates a DATA FIS and begins transmitting data to the device.
4. Hardware continues fetching PRDs as they become exhausted and fetching data from PRD locations until the transfer is complete. If the transfer is small enough, this data may fall under a single DATA FIS. The SATA Host controller will send FISes of maximum size to minimize FIS overhead on the data transfer.

Part IV - Hardware Actions – Command Wrap-Up

1. After the last piece of data has been accepted by the device, hardware awaits a device-to-host register FIS.
2. When the register FIS is received, hardware updates its task file shadow block.

Part V - Software Actions – Command Wrap-Up

1. Reading device status and BMIDE status

2. Complete request to OS
3. Error handling may occur, including device reset and DMA engine reinitialization.

29.5.1.2.2 ATA – Data from Device to Memory

The ATA Device-to-Memory command is the same as the ATA Memory-to-Device command, except that in Part III, hardware is receiving DATA FISes and writing data to memory instead of fetching data from memory and sending DATA FISes. The number of DATA FISes used is device-specific. Additionally, a DMA Activate FIS will not be received – the device will simply start sending a DATA FIS.

29.5.1.2.3 ATAPI

ATAPI DMA transfers are a combination of a PIO write command for the packet command transfer followed by a DMA command for the data transfer.

29.5.1.3 PIO Data Transfers

29.5.1.3.1 Write to Device

Part I - Software Actions – Command Start

Software writes to the task file to set up the command, with the final write being to the command register (1F7h, 177h).

Part II - Hardware Actions – Command Start

1. Upon seeing the command register written, hardware sends the register FIS to the device and awaits reception of either a PIO setup FIS or a device-to-host register FIS.
2. Upon reception of the PIO setup FIS (since the command was a PIO command), hardware updates the shadow block with the contents of the FIS and the current status, and holds the E-Status in reserve. If the “I” bit was set, it generates an interrupt.

Part III - Hardware/Software Actions – Data Transfer

Software writes to the data port as 16-bit quantities. Hardware assembles these writes into its FIFO and formulates a single DATA FIS to transmit the data. If software falls behind the data transmission rate of the interface (very likely), hardware will insert Idle characters.

Part IV - Hardware Actions – Command Wrap-Up

After the last piece of data has been accepted by the device, hardware updates the shadow block’s status register with the E-Status field.

Part V - Software Actions – Command Wrap-Up

1. Reading device status and BMIDE status.
2. Complete request to OS.
3. Error handling may occur, including device reset and DMA engine reinitialization.

29.5.1.3.2 Read from Device

The read ATA command is the same as the write ATA command except that in Part III, hardware is receiving the DATA FIS and writing data to memory instead of fetching data from memory and sending a DATA FIS.

29.5.1.4 Software-Assisted Queued DMA Transfer

In this mode of operation, software supports queuing, the SATA device supports queuing, but the SATA host controller does not. There are two general flows; one for a device that does not support the DMA setup FIS, and one for a device that does support the DMA setup FIS.

In both cases, software is doing the work to determine the tag of the transfer. No special hardware action is taken by the SATA host controller.

29.5.2 Error Handling

29.5.2.1 Errors on NSI

Errors on the memory interface will cause the behavior show in Table 958.

29.5.2.2 Errors on SATA Interface

Table 958. Memory Interface Error Results

Cycle Type	Address/Cmd Parity Error	Data Parity Error	Target Abort	Master Abort
I/O, Config Write	Set DPE bit Do not claim cycle	Set DPE bit Claim Cycle, data dropped Return completion success	NA	NA
I/O, Config Read	Set DPE bit Do not claim cycle	NA	NA	NA
Memory Write (to Intel® 3100 Chipset)	Not supported	NA	NA	NA
Memory Read (to Intel® 3100 Chipset)	NA	NA	NA	NA
Memory Write (from Intel® 3100 Chipset)	NA	NA	NA	MAI Not Supp
Memory Read (from Intel® 3100 Chipset)	NA	NA	NA	MAI Not Supp
I/O, Cfg Completion (read and write)	NA	NA	NA	MAI Not Supp
Mem. Read Comp. (from Intel® 3100 Chipset)	NA	NA	NA	MAI Not Supp
Mem. Read Comp. (to Intel® 3100 Chipset)	• Set DPE • Do not claim cycle • Set Error bit in bus master I/O space, offset 02h, bit 1 • Stop DMA engine • Needs a system reset to recover	Set DPE and DPD, Claim cycle • During PRD data transfer, abort DMA operation and set Error bit in bus master I/O space • During DMA data transfer, propagate the error to the device through crc error without setting Error Status bit.	Set RTA bit. Set Error bit and abort.	Set RMA bit. Set Error bit and abort.

There are several errors that can occur on the SATA interface, which may interrupt a data transfer.

There are two important aspects from the following tables that are important when validating the SATA host controller that must be followed:

- Errors that occur during DMA data in transfers (device sending data) that will result in data corruption will set the bus master error status bit (bit 1 of I/O offset 01h for primary, bit 1 of I/O offset 05h for secondary), while it may or may not set the bit

during other cases. If the SATA device generates R_ERR on DMA data out transfers (host sending data), the bus master error status bit is set, while it may or may not be set on other transfers.

- Errors that occur in PIO or DMA data in transfers (device generating data) that will result in data corruption will cause the SATA host controller to generate R_ERR on the SATA interface.

If the bus master error bit does get set for PIO transfers or non-data portions, it is acceptable. If the error bit does ***not*** get set for DMA data transfers, however, the SATA host controller is broken.

Table 959 breaks out conditions for the above rules. Additionally, other errors that may occur are listed for information purposes.

Table 959. Errors During Non-DATA FIS Reception

Error Type	Host Controller Behavior
Received Disparity Error/Illegal Character (K28.3)	Assume character is correct. Reset disparity counter. Do not set bus master error bit. Do not return R_ERR (still check for CRC errors).
Received Disparity Error/Illegal Character (D)	Return R_ERR at end of FIS. Do not set bus master error bit. (Device will retry)
Calculated different CRC than received or mal-formed FIS received[1]	Return R_ERR at end of FIS. Do not set bus master error bit. (Device will retry)
Phyready dropping unexpectedly	Send ALIGNs and return link FSM to IDLE. Do not set bus master error bit. (Device will retry)
Illegal FIS length for corresponding FIS type[2]	Return R_ERR at end of FIS. Do not set bus master error bit. (Device will retry).

Notes:

1. Mal-formed FIS = FIS not constructed according to link layer protocols.
2. Illegal length for corresponding FIS type. For example, the following FIS types and their corresponding lengths.
   - D2H Dma_Activate FIS length = 1 Dword
   - D2H Register FIS = 5 Dword
   - D2H Pio_Setup FIS = 5 Dword
   - D2H Set-Device-Bits = 2 Dword
   - D2H Bist FIS = 3 Dword

Table 960. Errors During PIO Data FIS Reception

Error Type	Host Controller Behavior
Received Disparity Error/ Illegal character (K28.3)	Assume character is correct. Reset disparity counter. Do not set bus master error bit. Do not return R_ERR (still check for CRC errors).
Received Disparity Error/Illegal Character (D)	Return R_ERR at end of FIS. Do not set bus master error bit. (Device will not retry.)
Calculated different CRC than received or mal-formed FIS received	Return R_ERR at end of FIS. Do not set bus master error bit. (Device will not retry.)
Phyready dropping unexpectedly	Send ALIGNs and return link FSM to IDLE. Do not set bus master error bit. (Device will not retry.)
Length of PIO Data FIS not matching transfer count in PIO_SETUP FIS	Return R_ERR at end of FIS. Do not set bus master error bit. (Device will not retry.)

Table 961. Errors During DMA Data FIS Reception

Error Type	Host Controller Behavior
Received Disparity Error/Illegal Character (K28.3)	Assume character is correct. Reset disparity counter. Do not set bus master error bit. Do not return R_ERR (still check for CRC errors).
Received Disparity Error/Illegal Character (D)	Return R_ERR at end of FIS. Set the bus master error bit. (Device will not retry.)
Calculated different CRC than received or mal-formed FIS received or internal buffer overflows (which could be caused by device violating HOLD-HOLDA latency)	Return R_ERR at end of FIS. Set the bus master error bit. (Device will not retry.)
Phyready dropping unexpectedly	Send ALIGNs and return link FSM to IDLE. Set the bus master error bit. (Device will not retry.)

Table 962. Errors During Unknown FIS type Reception

Error Type	Host Controller Behavior
Unknown FIS Type	The unknown FIS type is itself an error condition, and will result in bus master error bits being set and R_ERR being returned.

Note: Zero-length D2H Data FIS (i.e., Data FIS header immediately followed by CRC) and any D2H Data FIS following a D2H Dma_Activate FIS are treated as unknown FIS types for all purposes.

Table 963. Errors During FIS Transmission

Error Type	Host Controller Behavior
Non-data FIS: Received R_ERR (includes link protocol errors during transmission), or phyrdy dropping unexpectedly	Retry the FIS.
Data FIS: Received R_ERR (includes link protocol errors during transmission), or phyrdy dropping unexpectedly	Set the bus master error bit. Do not retry the FIS.

Notes:
1. Malformed FIS = FIS not constructed according to link layer protocols
2. Illegal length for corresponding FIS type. For example, the following FIS types and their corresponding lengths:
   1. D2H Dma_Activate FIS length = 1 Dword
   2. D2H Register FIS = 5 Dword
   3. D2H Pio_Setup FIS = 5 Dword
   4. D2H Set-Device-Bits = 2 Dword
   5. D2H Bist FIS = 3 Dword
3. Zero-length D2H Data FIS (i.e., Data FIS header immediately followed by CRC), and any D2H Data FIS following a D2H Dma_Activate FIS are treated as unknown FIS types for all purposes.

29.5.3 Hot Plug Operation

Dynamic Hot-Plug (e.g., surprise removal) is not supported in legacy modeby the SATA host controller without special support from AHCI and the proper board hardware. However, basic SATA swap bay is supported using the PSC register configuration bits and power management flows. A device can be powered down by software and the port can then be disabled, allowing removal and insertion of a new device.

Note: This SATA swap bay operation requires board hardware (implementation specific), BIOS, and operating system support.

29.5.4 48-Bit ("Large") LBA Operation Requirements

The SATA host controller supports 48-bit LBA through the host-to-device register FIS when accesses are performed via writes to the task file. The SATA host controller will ensure that the correct data is put into the correct byte of the host-to-device FIS as follows: if only one write is performed, the data goes to the location specified; if a second write is performed, the data from the first write is shifted into the "upper" location, and the data from the second write goes to the location specified.

Suppose a sequence of writes occurred to the taskfile as follows:

- 1f2h (Sector Count) ..... - 21h
- 1f2h (Sector Count) ..... - 22h
- 1f3h (Sector Number) ..... - 31h
- 1f3h (Sector Number) ..... - 32h
- 1f4h (Cylinder Low) ..... - 41h
- 1f4h (Cylinder Low) ..... - 42h
- 1f5h (Cylinder High) ..... - 51h
- 1f5h (Cylinder High) ..... - 52h

The resulting FIS when the command or control register is written will have the values shown in Table 964.

Table 964. FIS Values

Dword					
0	Features	Command	C	0000000	FIS Type (27h)
1	Dev / Head	Cyl High (52h)	Cyl Low (42h)		Sector Number (32h)
2	Features (exp)	Cyl High Exp (51h)	Cyl Low Exp (41h)		Sector Num Exp (31h)
3	Control	Reserved (0)	Sector Count Exp (21h)		Sector Count (22h)
4	Reserved (0)	Reserved (0)	Reserved (0)		Reserved (0)

The writes do not have to come in any specific order. All that is necessary is that the writes to these registers act as a FIFO – the second write moves data from the first write into a new location.

There are also special considerations when reading from the task file to support 48-bit LBA operation. Software may need to read all 16 bits. Since the registers are only 8 bits wide and act as a FIFO, a bit must be set in the device/control register, which is at offset 3f6h for primary and 376h for secondary (or their native counterparts).

If software clears bit 7 of the control register before performing a read, the last item written is returned from the FIFO. If software sets bit 7 of the control register before performing a read, the first item written is returned from the FIFO.

29.5.5 Power Management Operation

29.5.5.1 Introduction

Power management of the SATA Controller and ports will cover operations of the host controller and the SATA wire. This specification does not cover any power management that a SATA device may do internally that is transparent to the interface.

29.5.5.2 Power State Mappings

- The *PCI Specification* defines power management states for devices, which is applied to the SATA host controller. They are:
- D0 – working (required)
- D1 – light sleep (not supported)
- D2 – deeper sleep (not supported)
- D3 – very deep sleep (required). This state is split into two sub-states, $D3_{HOT}$ (can respond to PCI configuration accesses) and $D3_{COLD}$ (cannot respond to PCI configuration accesses). These two sub-states are considered the same, where $D3_{HOT}$ has V_{CC}, but $D3_{COLD}$ does not. This is the only state allowed for the host controller when the system is in an S3 and S5 state.

SATA devices may also have multiple power states. From parallel ATA, three device states are supported through ACPI. They are:

- D0 – Device is working and instantly available.
- D1 – Device enters when it receives a STANDBY IMMEDIATE command. Exit latency from this state is in seconds
- D3 – From the SATA device's perspective, no different than a D1 state in that it is entered via the STANDBY IMMEDIATE command. However, an ACPI method is also called, which resets the device and then cut its power through. (In IICH, this included setting the tri-state bits of the interface, a GPIO which reset the device, and a GPIO to cut power to that device.

Each of these device states are subsets of the host controller's D0 state. This is partially because host controllers (as integrated in Intel components) have not supported host-based power management, and also because the device must be put into one of the lower power states before power could be removed from the host.

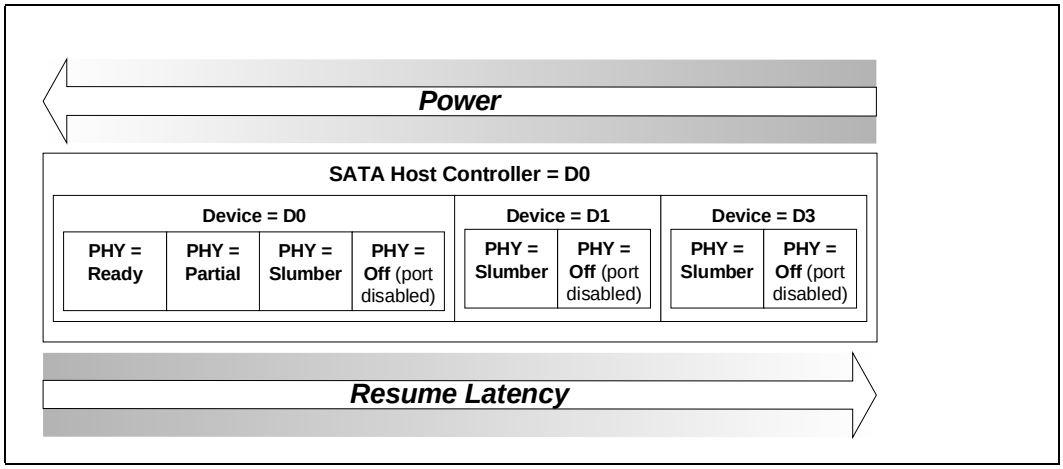
Finally, SATA defines three PHY layer power states, which have no equivalent mappings to parallel ATA. They are:

- PHY READY – PHY logic and PLL are both on and active,
- Partial – PHY logic is powered, but in a reduced state. Exit latency is no longer than 10 µs,
- Slumber – PHY logic is powered, but in a reduced state. Exit latency can be up to 10 ms.

Since these states have much lower exit latency than the ACPI D1 and D3 states, the SATA Controller defines these states as sub-states of the device D0 state.

The following picture gives a hierarchical view of power states of SATA.

Figure 86. SATA Power States



29.5.5.3 Power State Transitions

Transitioning between various states is initiated by different levels of software and hardware.

29.5.5.3.1 Partial and Slumber State Entry/Exit

The partial and slumber states save interface power when the interface is idle.The SATA Controller defines PHY layer power management (as performed via primitives) as a driver operation from the host side, and a device mechanism on the device side. The SATA Controller will accept device transition types, but will not issue any transitions as a host. All received requests from an SATA device is ACKed.

When an operation is performed to the SATA Controller such that it needs to use the SATA cable, the controller must check whether the link is in the Partial or Slumber states, and if so, must issue a COM_WAKE to bring the link back online. Similarly, the SATA device must perform the same action.

29.5.5.3.2 Device D1, D3 States

These states are entered after some period of time when software has determined that no commands are sent to this device for some time. The mechanism for putting a device in these states does not involve any work on the host controller, other then sending commands over the interface to the device. The command most likely to be used in ATA/ATAPI is the “STANDBY IMMEDIATE” command.

29.5.5.3.3 Host Controller D3 state

After the interface and device have been put into a low power state, the host controller may be put into a low power state. This is performed via the PCI power management registers in configuration space.

There are two very important aspects to note when using PCI power management.

1. When the power state is D3, only accesses to configuration space are allowed. Any attempt to access the memory or I/O spaces must result in master abort.
2. When the power state is D3, no interrupts may be generated, even if they are enabled. If an interrupt status bit is pending when the controller transitions to D0, an interrupt may be generated.

When the controller is put into D3, it is assumed that software has properly shut down the device and disabled the ports. Therefore, there is no need to sustain any values on the port wires. The interface is treated as if no device is present on the cable, and power is minimized.

When returning from a D3 state, an internal reset will not be performed.

29.5.5.4 Non-AHCI Mode PME# Generation

When in legacy (non-AHCI mode of operation), the SATA controller does not generate PME#. This includes attach events (since the port must be disabled).

29.5.5.5 SMI Trapping (APM)

The ATC register in configuration space contains control for generating SMI# on accesses to the IDE I/O spaces. These bits map to the legacy ranges only (1F0h-1F7h, 3F4h-3F6h, 170h-177h, and 374h-376h). Trapping will not occur on the native IDE ranges defined by PCMDBA, PCTLBA, SCMDBA, SCTLBA, or LBAR. If the SATA controller is in legacy mode and is using these addresses, accesses to one of these ranges with the appropriate bit set will cause the cycle to not be forwarded to the SATA controller and SMI# to be generated.

SMI trapping is specifically supported in the following configurations:

- PATA controller disabled, and SATA controller in legacy addressing mode (non-combined)
- SATA controller disabled, and PATA controller in legacy addressing mode (non-combined)
- Combined mode, legacy addressing

Additionally, an ATS register bit will get set on an access to these ranges if the corresponding bit in the ATC register is set.

29.5.6 SATA Interrupts

The following table summarizes interrupt behavior for MSI and wire modes. In Table 965, “bits” refers to the four possible interrupt bits in I/O space, which are:

- PSTS.PRDIS (offset 02h, bit 7)
- PSTS.I (offset 02h, bit 2)
- SSTS.PRDIS (offset 0Ah, bit 7)
- SSTS.I (offset 0Ah, bit 2)

Table 965. MSI vs. PCI IRQ Actions

Interrupt Register	Wire-Mode Action	MSI Action
All bits ‘0’	Wire inactive	No action
One or more bits set to ‘1’	Wire active	Send message
One or more bits set to ‘1’, new bit gets set to ‘1’	Wire active	Send message
One or more bits set to ‘1’, software clears some (but not all) bits	Wire active	Send message
One or more bits set to ‘1’, software clears all bits	Wire inactive	No action
Software clears one or more bits, and one or more bits are set on the same clock.	Wire active	Send message

29.5.7 Staggered Spin-up

To support staggered spin-up with legacy software, the AHCI memory space register CAP.SSS must be cleared, and the configuration register PCS is used to enable/disable the port.

29.5.8 Hardware/Software Operation for Detecting SATA Device Presence

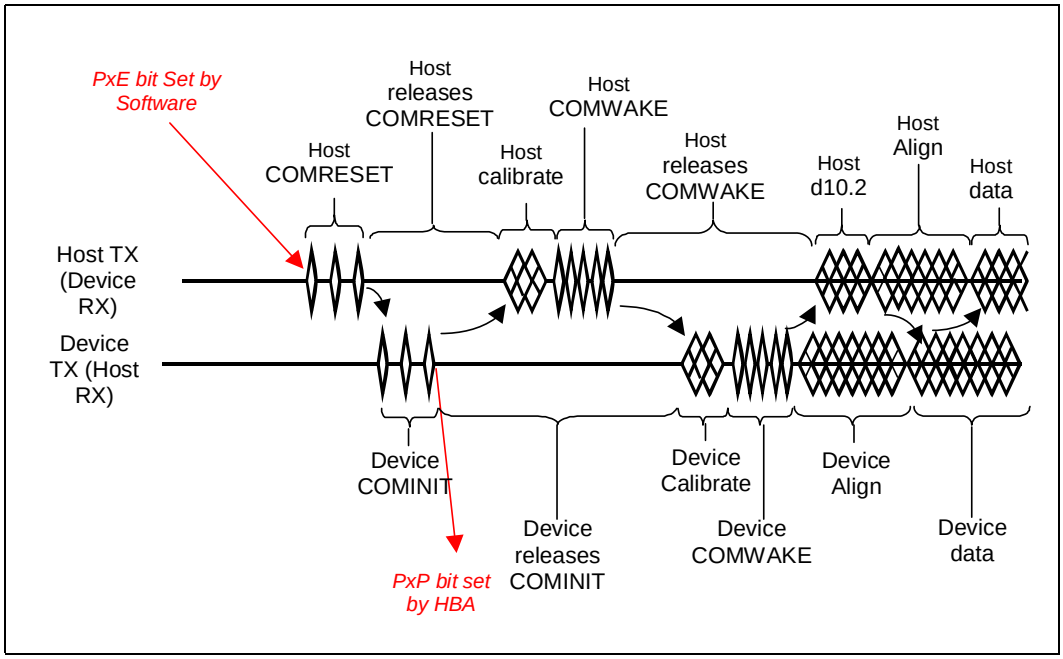
29.5.8.1 Introduction

In legacy mode, the SATA controller does not generate interrupts based on hot plug/unplug events. However, the SATA PHY does know when a device is connected (if not in a partial or slumber state), and it is beneficial to communicate this information to host software as this will greatly reduce boot times and resume times.

29.5.8.2 Hardware Flow

The flow for using these bits is shown in Figure 87. The 'PxE' bit refers to PCS.P0E, PCS.P1E, PCS.P2E, and PCS.P3E bits, depending on the port being checked, and the 'PxP' bit refers to the PCS.P0P, PCS.P1P, PCS.P2P, and PCS.P3P bits, depending on the port being checked.

Figure 87. Hardware Flow for Port Enable/Device Present Bits



Note: The SATA host controller's COMRESET length is six data bursts.

29.5.8.3 Software Flow

The software flow is as follows:

- Sometime after power-on reset or resume from suspend, software will set the PCS.PxE bits, depending on which ports it wants to scan. They can both be set together.

- The *Serial ATA Specification* indicates that the COMRESET sequence to device detection is to be 880 µs. For best results, software must wait some period longer than this, such as 10 ms.
- Software will read the PCS.PxP bits. If the bit is set, a device is present. If the bit is cleared, a device is not present.

If a port was disabled, software checks to see if a new device is connected by periodically reenabling the port, and waiting 2-3 ms to see if a device is present. If one is not, it can disable the port and check again later. If a port remains enabled, software can periodically poll PCS.PxP to see if a new device is connected.

29.5.9 SATA LED

The SATA_LED# output is driven whenever the BSY bit is set in any SATA port. The SATA_LED# is an active-low open drain output. When SATA_LED# is low, the LED is active. When SATA_LED# is tri-stated, the LED is inactive.When SATA_LED# is low, the LED must be active. When SATA_LED# is high, the LED must be inactive.

29.5.10 Staggered Spin-up Support

To support staggered spin-up in legacy operation, BIOS software individually cycles through the PCS.PxE bits in configuration space. It sets one bit and waits for the device to not be busy (as indicated by the status register). It then moves onto the next port.

29.6 AHCI Operation

The Intel® 3100 Chipset provides hardware support for Advanced Host Controller Interface (AHCI), a new programming interface for SATA host controllers developed through a joint industry effort. AHCI defines transactions between Intel® 3100 Chipset's SATA controller and software and enables advanced performance and usability with SATA. Platforms supporting AHCI may take advantage of performance features such as no master/slave designation for SATA devices—each device is treated as a master—and hardware assisted native command queuing. AHCI also provides usability enhancements such as Hot-Plug. AHCI requires appropriate software support (e.g., an AHCI driver) and for some features, hardware support in the SATA device or additional platform hardware.

The Intel® 3100 Chipset supports all of the mandatory features of the *AHCI Specification, Rev 1.0.*

For reliable device removal notification while in AHCI operation without the use of interlock switches (surprise removal), interface power management must be disabled for the associated port. See Section 7.3.1 of the *AHCI Specification* for more information.

29.6.1 System Memory Structures

The SATA controller supports six ports, and each port supports 32 commands. The command list and received FIS may live in 64-bit space.

29.6.2 Hot Plug Operation

Figure 88. Port System Memory Structure

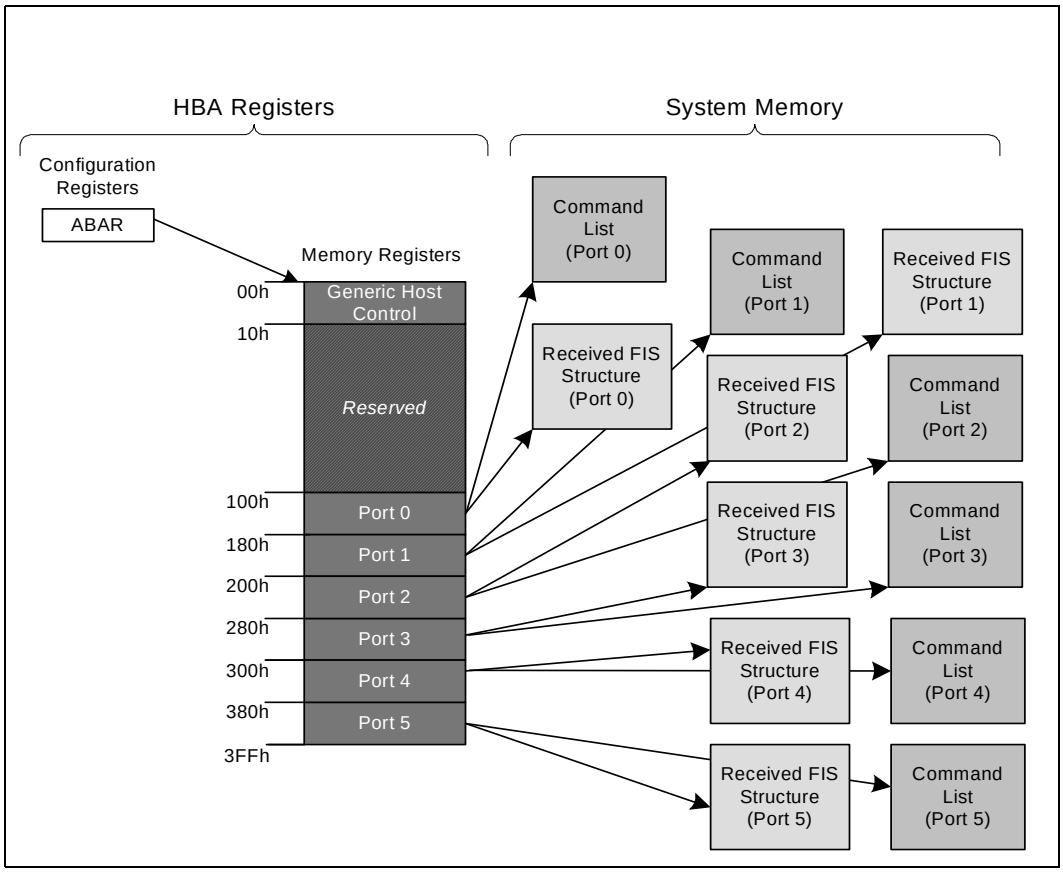
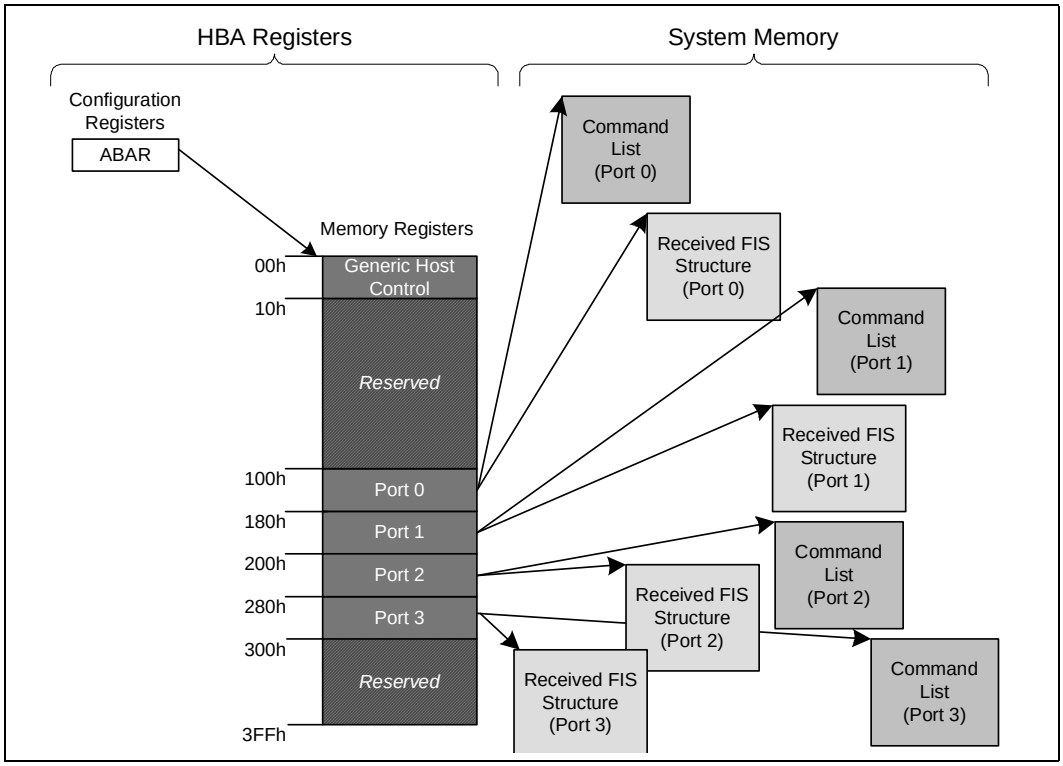


Figure 89. Port System Memory Structure



Refer to Chapter 7 of the *AHCI Specification* for details. The Intel® 3100 Chipset supports Hot Plug Surprise Removal Notification. However Hot Plug Surprise Removal Notification (without an interlock switch) is mutually exclusive with the PARTIAL and SLUMBER power management states. The following conceptual flows describe the different software initialization steps necessary to support Surprise Removal Notification or Power Management.

29.6.2.1 Per Port Software Initialization to Support Surprise Insertion and Removal Notification:

1. Set PxSCTL.IPM to 3h to disable both PARTIAL and SLUMBER transitions invoked by the drive.
2. Set PxCMD.ALPE to 0 to disable aggressive power management by the HBA.
3. Set PxIE.PCE to 1 to enable interrupts for hot plug insertions.
4. The PxIS.PCS interrupt status bit indicates to software that a hot plug insertion event occurred.
5. Write a 1 to the PxSERR.DIAG.X bit to clear it and the PxIS.PCS bit.

After a drive is detected on the port:

6. Set the newly defined PxIE.PRCE to 1 to enable interrupts for hot plug removals.
7. Issue the Set Features command to the drive to disable interface power management.
8. The newly defined PxIS.PRCS interrupt status bit indicates to software that a hot plug removal event occurred.

9. Set the PxIE.PRCE to 0 to disable interrupts so that only the PxIS.PCS bit will generate an insertion event interrupt. If this bit is left enabled two interrupts may be generated on the insertion event.

29.6.2.2 Per Port Software Initialization to Support PARTIAL and SLUMBER Power Management:

1. Set PxSCTL.IPM to 0h to enable PARTIAL and SLUMBER transitions for the HBA port.
2. Set PxCMD.ALPE and PxCMD.ASP appropriately based on aggressive power management policy.
3. Set PxIE.PRCE to 0 to disable interrupts when PhyRdy changes state due to PARTIAL and SLUMBER transitions.
4. During active operation PxSSTS.DET = 3h indicates to software that the link is operational.

29.6.3 Power Management Operation

Refer to Chapter 8 of the *AHCI Specification* for details.

The

30.0 Device 28, Functions 0, 1, 2, 3: PCI Express* Root Ports B

30.1 Overview

This set of registers controls an individual PCI Express root port, Port B (PEB[3:0]). There are four ports in the IICH. These are all in device 28 and functions 0–3. Port B0 (PEB0) is function 0, Port B1 (PEB1) is function 1, Port B2 (PEB2) is function 2, and Port B3 (PEB3) is function 3.

30.2 Configuration Registers

Reserved registers are read-only and return all zeros.

Start	End	Register Block Description
000	03F	PCI Header
040	06F	PCI Express Root Port Capability
080	08F	Message Signaled Interrupt Capability
090	09F	Subsystem Vendor Capability
0A0	0AF	PCI Power Management Capability
0D0	0DF	Port Configuration
0F8	0F8	Manufacturer's ID
100	13F	VC Configuration
140	17F	Advanced Error Reporting Configuration
180	18F	Root Complex Topology Capability Structure
300	3FF	DFT Configuration (see DFT section)

30.2.1 PCI Header

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 966. Register Summary: PCI Header Registers

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
00h	03h	ID	Identification Registers	See Desc8086h	RO
04h	05h	CMD	Command Register	0000h	RW, RO
06h	07h	STS	Device Status Register	00100h	RWC, RO
08h	08h	RID	Revision ID Register	See Desc	RO
09h	0Bh	CC	Class Code Register	060400h	RO
0Ch	0Ch	CLS	Cache Line Size Register	00h	RW
0Dh	0Dh	PLT	Primary Latency Timer Register	00h	RO
0Eh	0Eh	HTYPE	Header Type Register	81h	RO
18h	1Ah	BNUM	Bus Number Register	000000h	RW
1Bh	1Bh	SLT	Secondary Latency Timer Register	see desc	see desc
1Ch	1Dh	IOBL	I/O Base and Limit Register	0000h	RO, RW
1Eh	1Fh	SSTS	Secondary Status Register	0000h	RWC, RO
20h	23h	MBL	Memory Base and Limit Register	00000000h	RW, RO
24h	27h	PMBL	Prefetchable Memory Base and Limit Register	00010001h	RO, RW
28h	2Bh	PMBU32	Prefetchable Memory Base Upper 32-bits Register	00000000h	RW
2Ch	2Fh	PMLU32	Prefetchable Memory Limit Upper 32-bits Register	00000000h	RW
34h	34h	CAPP	Capability List Pointer Register	40h	RO
3Ch	3Dh	INTR	Interrupt Information Register	xx00h	RO, RW
3Eh	3Fh	BCTRL	Bridge Control Register	0000h	RO, RW

30.2.1.1 Offset 00 - 03h: VID and DID – Identification Registers

Table 967. Offset 00 - 03h: VID and DID – Identification Registers

Device: 28 ***Function:*** 0/1
Offset: 00 - 03h ***Size:*** *32 bit*
Default Value: *See Desc8086h* ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:16	DID	**Device Identification:** This field is defined by the following table. Port # / Device ID (FDPCIE=0) / DeviceID (FDPCIE=1) 1 / 2690h / 2691h 2 / 2692h / 2693h 3 / 2694h / 2695h 4 / 2696h / 2697h	See Desc	RO
15:00	VID	**Vendor Identification:** Indicates Intel as the manufacturer.	8086h	RO

30.2.1.2 Offset 04 - 05h: CMD – Device Command Register

Table 968. Offset 04 - 05h: CMD – Device Command Register

Device: 28	***Function:*** 0,1,2,3
Offset: 04 - 05h	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:11	Reserved	Reserved	0	
10	ID	**Interrupt Disable:** This disables pin-based INTx# interrupts on enabled hot plug and power management events. This bit has no effect on MSI operation. 0 = Internal INTx# messages are generated if there is an interrupt for hot plug or power management and MSI is not enabled. 1 = Internal INTx# messages are not generated. This bit does not affect interrupt forwarding from devices connected to the root port. Assert_INTx and Deassert_INTx messages are still forwarded to the internal interrupt controllers if this bit is set.	0	RW
09	FBE	**Fast Back to Back Enable:** Reserved per the *PCI Express* Base Specification*.	0	
08	SEE	**SERR# Enable:** 0 = Disable. 1 = Enables the root port to generate an SERR# message when PSTS.SSE is set.	0	RW
07	WCC	**Wait Cycle Control:** Reserved per the *PCI Express* Base Specification*.	0	
06	PERE	**Parity Error Response Enable:** 0 = Disable. 1 = Indicates that the device is capable of reporting parity errors as a master.	0	RW
05	VGA_PSE	**VGA Palette Snoop:** Reserved per the *PCI Express* Base Specification*.	0	
04	MWIE	**Memory Write and Invalidate Enable:** Reserved per the *PCI Express* Base Specification*.	0	
03	SCE	**Special Cycle Enable:** Reserved per the *PCI Express* Base Specification*.	0	
02	BME	**Bus Master Enable:** 0 = All cycles from the device are master aborted. 1 = Allows the root port to forward cycles from a PCI Express device.	0	RW
01	MSE	**Memory Space Enable:** 0 = These memory cycles are master aborted. 1 = Memory cycles within the range specified by the memory base and limit registers can be forwarded to the PCI Express device.	0	RW
00	IOSE	**I/O Space Enable:** 0 = These cycles are master aborted. I/O cycles within the range specified by the I/O base and limit registers are master aborted. 1 = I/O cycles within the range specified by the I/O base and limit registers can be forwarded to the PCI Express device.	0	RW

30.2.1.3 Offset 06 - 07h: PSTS – Primary Status Register

Table 969. Offset 06 - 07h: PSTS – Primary Status Register

Device: 28	***Function:*** 0,1,2,3
Offset: 06 - 07h	***Size:*** *16 bit*
Default Value: *0010h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15	DPE	**Detected Parity Error:** 0 = No parity error detected. 1 = Set when the root port receives a command or data with a parity error. This is set even if CMD.PERE (Section 30.2.1.2) is not set.	0	RWC
14	SSE	**Signaled System Error:** 0 = No system error signaled. 1 = Set when the root port signals a system error to the internal SERR# logic.	0	RWC
13	RMA	**Received Master Abort:** 0 = Root port has not received a completion with unsupported request status. 1 = Set when the root port receives a completion with unsupported request status.	0	RWC
12	RTA	**Received Target Abort:** 0 = Root port has not received a completion with completer abort. 1 = Set when the root port receives a completion with completer abort.	0	RWC
11	STA	**Signaled Target Abort:** 0 = No target abort received. 1 = Set whenever the root port forwards a target abort received from the downstream device.	0	RWC
10:09	PDTS	**Primary DEVSEL# Timing Status:** Reserved per the *PCI Express* Base Specification*.	00	
08	DPD	**Master Data Parity Error Detected:** 0 = No data parity error received. 1 = Set when the root port receives a completion with a data parity error and CMD.PERE (Section 30.2.1.2) is set.	0	RWC
07	PFBC	**Primary Fast Back to Back Capable:** Reserved per the *PCI Express* Base Specification*.	0	
06	Reserved	Reserved	0	
05	PC66	**Primary 66 MHz Capable:** Reserved per the *PCI Express* Base Specification*.	0	
04	CLIST	**Capabilities List:** Indicates the presence of a capabilities list, hardwired to 1.	1	RO
03	IS	**Interrupt Status:** Indicates status of hot plug and power management interrupts on the root port that result in INTx# message generation. 0 = Interrupt is deasserted. 1 = Interrupt is asserted. ***Note:*** This bit is not set if MSI is enabled. If MSI is not enabled, this bit is set regardless of the state of CMD.ID.	0	RO
02:00	Reserved	Reserved	000	

30.2.1.4 Offset 08h: RID – Revision ID Register

Table 970. Offset 08h: RID – Revision ID Register

Device: 28	***Function:*** 0,1,2,3
Offset: 08h	***Size:*** *8 bit*
Default Value: *See Desc*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	RID	**Revision ID:** Indicates the revision of the bridge. The value of this register always reflects the value of the LPC Interface Revision ID (Dev 31 Func 0, offset 08h).	See Desc	RO

30.2.1.5 Offset 09 - 0Bh: CC – Class Code Register

Table 971. Offset 09 - 0Bh: CC – Class Code Register

Device: 28	***Function:*** 0,1,2,3
Offset: 09h - 0Bh	***Size:*** *24 bit*
Default Value: *060400h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
23:16	BCC	**Base Class Code:** Indicates the device is a bridge device.	06h	RO
15:08	SCC	**Sub-Class Code:** Indicates the device is a PCI-to-PCI bridge.	04h	RO
07:00	PI	**Programming Interface:** No special programming interface for a bridge.	00h	RO

30.2.1.6 Offset 0Ch: CLS – Cache Line Size Register

Table 972. Offset 0Ch: CLS – Cache Line Size Register

Device: 28	***Function:*** 0,1,2,3
Offset: 0Ch	***Size:*** *8 bit*
Default Value: *00h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	LS	**Line Size:** This is read/write but contains no functionality, per *PCI Express* Base Specification*.	0h	RW

30.2.1.7 Offset 0Dh: PLT – Primary Latency Timer Register

Table 973. Offset 0Dh: PLT – Primary Latency Timer Register

Device: 28	***Function:*** 0,1,2,3
Offset: 0Dh	***Size:*** *8 bit*
Default Value: *00h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:03	CT	**Latency Count.** Reserved per the *PCI Express* Base Specification*.	0h	
02:00	Reserved	Reserved	0h	

30.2.1.8 Offset 0Eh: HTYPE – Header Type Register

Table 974. Offset 0Eh: HTYPE – Header Type Register

Device: 28	***Function:*** 0,1,2,3
Offset: 0Eh	***Size:*** *8 bit*
Default Value: *81h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07	MFD	**Multi-Function Device:** 0 = Single-function device. 1 = Multi-function device.	1	RO
06:00	HTYPE	**Header Type:** Identifies the header layout of the configuration space, which is a PCI-to-PCI bridge. Hardwired to 01h.	01h	RO

30.2.1.9 Offset 18 - 1Ah: BNUM – Bus Numbers Register

Table 975. Offset 18 - 1Ah: BNUM – Bus Numbers Register

Device: 28	***Function:*** 0,1,2,3
Offset: 18 - 1Ah	***Size:*** *24 bit*
Default Value: *000000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
23:16	SBBN	**Subordinate Bus Number:** Indicates the highest PCI bus number below the bridge.	00h	RW
15:08	SCBN	**Secondary Bus Number:** Indicates the bus number of the port.	00h	RW
07:00	PBN	**Primary Bus Number:** Indicates the bus number.	00h	RW

30.2.1.10 Offset 1Bh: SLT – Secondary Latency Timer Register

This register is reserved for a root port as per the *PCI Express* Specification*.

30.2.1.11 Offset 1C - 1Dh: IOBL – I/O Base and Limit Register

Table 976. Offset 1C - 1Dh: IOBL – I/O Base and Limit Register

Device: 28	***Function:*** 0,1,2,3
Offset: 1C - 1Dh	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:12	IOLA	**I/O Address Limit:** I/O Base bits corresponding to address lines 15:12 for 4 Kbyte alignment. Bits 11:00 are assumed to be padded to FFFh.	0h	RW
11:08	IOLC	**I/O Limit Address Capability:** Indicates that the bridge does not support 32-bit I/O addressing.	0h	RO
07:04	IOBA	**I/O Base Address:** I/O Base bits corresponding to address lines 15:12 for 4 Kbyte alignment. Bits 11:00 are assumed to be padded to 000h.	0h	RW
03:00	IOBC	**I/O Base Address Capability:** Indicates that the bridge does not support 32-bit I/O addressing.	0h	RO

30.2.1.12 Offset 1E - 1Fh: SSTS – Secondary Status Register

Table 977. Offset 1E - 1Fh: SSTS – Secondary Status Register (Sheet 1 of 2)

Device: 28	***Function:*** 0,1,2,3
Offset: 1E - 1Fh	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15	DPE	**Detected Parity Error:** 0 = No error. 1 = The port received a poisoned TLP.	0	RWC
14	RSE	**Received System Error:** 0 = No error. 1 = The port received an ERR_FATAL or ERR_NONFATAL message from the device.	0	RWC
13	RMA	**Received Master Abort:** 0 = Unsupported Request not received. 1 = The port received a completion with “Unsupported Request” status from the device. In addition if a device is not connected to the PCI express port (PCI express link is in link down state) when a config cycle is sent to the secondary bus, a master abort is generated and this bit is set.	0	RWC
12	RTA	**Received Target Abort (RTA):** 0 = Completion Abort not received. 1 = The port received a completion with “Completion Abort” status from the device.	0	RWC
11	STA	**Signaled Target Abort:** 0 = Completion Abort not sent. 1 = The port generated a completion with “Completion Abort” status to the device.	0	RWC
10:09	SDTS	**Secondary DEVSEL# Timing Status:** Reserved per *PCI Express* Base Specification*.	00	

Table 977. Offset 1E - 1Fh: SSTS – Secondary Status Register (Sheet 2 of 2)

Device: 28	***Function:*** 0,1,2,3
Offset: 1E - 1Fh	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
08	DPD	**Data Parity Error Detected (DPD):** 0 = Conditions below did not occur. 1 = Set when the BCTRL.PERE (D28, FO/F1, 3E, bit 0) is set, and either of the following two conditions occurs: • Port receives completion marked poisoned. • Port poisons a write request to the secondary side.	0	RWC
07	SFBC	**Secondary Fast Back-to-Back Capable:** Reserved per the *PCI Express* Base Specification*.	0	
06	Reserved	Reserved	0	
05	SC66	**Secondary 66 MHz Capable:** Reserved per the *PCI Express* Base Specification*.	0	
04:00	Reserved	Reserved	0	

30.2.1.13 Offset 20 - 23h: MBL – Memory Base and Limit Register

Accesses that are within the ranges specified in this register are sent to the attached device if CMD.MSE is set. Accesses from the attached device that are outside the ranges specified are forwarded if CMD.BME is set. The comparison performed is MB >= AD[31:20] <= ML.

Table 978. Offset 20 - 23h: MBL – Memory Base and Limit Register

Device: 28	***Function:*** 0,1,2,3
Offset: 20 - 23h	***Size:*** *32 bit*
Default Value: *00000000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:20	ML	**Memory Limit:** These bits are compared with bits 31:20 of the incoming address to determine the upper 1 Mbyte aligned value of the range.	000h	RW
19:16	Reserved	Reserved	0h	
15:04	MB	**Memory Base:** These bits are compared with bits 31:20 of the incoming address to determine the lower 1 Mbyte aligned value of the range.	000h	RW
03:00	Reserved	Reserved	0h	

30.2.1.14 Offset 24 - 27h: PMBL – Prefetchable Memory Base and Limit Register

Accesses that are within the ranges specified in this register are sent to the device if CMD.MSE (Section 30.2.1.2) is set. Accesses from the device that are outside the ranges specified are forwarded if CMD.BME (Section 30.2.1.2) is set. The comparison performed is PMBU32:PMB >= AD[63:32]:AD[31:20] <= PMLU32:PML.

Table 979. Offset 24 - 27h: PMBL – Prefetchable Memory Base and Limit Register

Device: 28	***Function:*** 0,1,2,3
Offset: 24 - 27h	***Size:*** *32 bit*
Default Value: *00010001h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:20	PML	**Prefetchable Memory Limit:** These bits are compared with bits 31:20 of the incoming address to determine the upper 1 Mbyte aligned value of the range.	000h	RW
19:16	I64L	**64-bit Indicator:** Indicates support for 64-bit addressing.	1h	RO
15:04	PMB	**Prefetchable Memory Base:** These bits are compared with bits 31:20 of the incoming address to determine the lower 1 Mbyte aligned value (inclusive) of the range. The incoming address must be greater than or equal to this value.	000h	RW
03:00	I64B	**64-bit Indicator:** Indicates support for 64-bit addressing.	1h	RO

30.2.1.15 Offset 28 - 2Bh: PMBU32 – Prefetchable Memory Base Upper 32-Bits Register

Table 980. Offset 28 - 2Bh: PMBU32 – Prefetchable Memory Base Upper 32-Bits Register

Device: 28	***Function:*** 0,1,2,3
Offset: 28 - 2Bh	***Size:*** *32 bit*
Default Value: *00000000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:00	PMBU	**Prefetchable Memory Base Upper Portion:** Upper 32 bits of the prefetchable address base.	00000000h	RW

30.2.1.16 Offset 2C - 2Fh: PMLU32 – Prefetchable Memory Limit Upper 32-Bits Register

Table 981. Offset 2C - 2Fh: PMLU32 – Prefetchable Memory Limit Upper 32-Bits Register

Device: 28	***Function:*** 0,1,2,3
Offset: 2C - 2Fh	***Size:*** *32 bit*
Default Value: *00000000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:00	PMLU	**Prefetchable Memory Limit Upper Portion:** Upper 32 bits of the prefetchable address limit.	00000000h	RW

30.2.1.17 Offset 34h: CAPP – Capabilities List Pointer Register

Table 982. Offset 34h: CAPP – Capabilities List Pointer Register

Device: 28	***Function:*** 0,1,2,3
Offset: 34h	***Size:*** *8 bit*
Default Value: *40h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
07:00	PTR	**Capabilities Pointer:** Indicates that the pointer for the first entry in the capabilities list is at 40h in configuration space.	40h	RO

30.2.1.18 Offset 3C - 3Dh: INTR – Interrupt Information Register

Table 983. Offset 3C - 3Dh: INTR – Interrupt Information Register

Device: 28	***Function:*** 0,1,2,3
Offset: 3C - 3Dh	***Size:*** *16 bit*
Default Value: *XX00h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:08	IPIN	**Interrupt Pin:** Indicates the interrupt pin driven by the root port. At reset, this register takes on the following values, which reflect the reset state of the D28IP register in the configuration space: Port Bits[15:12] Bits[11:08] 1 0h D28IP.P1IP 2 0h D28IP.P2IP 3 0h D28IP.P3IP 4 0h D28IP.P4IP The value that is programmed into D28IP is always reflected in this register.	See Table	RO
07:00	ILINE	**Interrupt Line:** Software written value to indicate which interrupt line (vector) the interrupt is connected to. No hardware action is taken on this register.	00h	RW

30.2.1.19 Offset 3E - 3Fh: BCTRL – Bridge Control Register

Table 984. Offset 3E - 3Fh: BCTRL – Bridge Control Register

Device: 28	***Function:*** 0,1,2,3
Offset: 3E - 3Fh	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:12	Reserved	Reserved	0000	
11	DTSE	**Discard Timer SERR# Enable:** Reserved per *PCI Express* Base Specification*.	0	
10	DTS	**Discard Timer Status:** Reserved per *PCI Express* Base Specification*.	0	
09	SDT	**Secondary Discard Timer:** Reserved per *PCI Express* Base Specification*.	0	
08	PDT	**Primary Discard Timer:** Reserved per *PCI Express* Base Specification*.	0	
07	FBE	**Fast Back to Back Enable:** Reserved per *PCI Express* Base Specification*.	0	
06	SBR	**Secondary Bus Reset:** Triggers a Hot Reset on the PCI Express port.	0	RW
05	MAM	**Master Abort Mode:** Reserved per *PCI Express* Base Specification*.	0	
04	V16	**VGA 16-Bit Decode:** 0 = VGA range is enabled. 1 = The I/O aliases of the VGA range (see BCTRL:VE definition below) are not enabled, and only the base I/O ranges can be decoded.	0	RW
03	VE	**VGA Enable:** 0 = The ranges below are not claimed off by the root port. 1 = The following ranges are claimed off by the root port: • Memory ranges A0000h– BFFFFh • I/O ranges 3B0h–3BBh and 3C0h–3DFh, and all aliases of bits 15:10 in any combination of ones.	0	RW
02	IE	**ISA Enable:** This bit only applies to I/O addresses that are enabled by the I/O Base and I/O Limit registers and are in the first 64 KByte of PCI I/O space. 0 = The root port does not block any forwarding as described below. 1 = The root port blocks any forwarding to the device of I/O transactions addressing the last 768 bytes in each 1 KByte block (offsets 100h to 3FFh).	0	RW
01	SE	**SERR# Enable:** 0 = The messages described below are not forwarded. 1 = ERR_COR, ERR_NONFATAL, and ERR_FATAL messages received are forwarded.	0	RW
00	PERE	**Parity Error Response Enable:** 0 = Poisoned write TLPs and completions indicating poisoned TLPs do not set the SSTS.DPD. 1 = Poisoned write TLPs and completions indicating poisoned TLPs sets the SSTS.DPD.	0	RW

30.2.2 Root Port Capability Structure

The registers follow the PCI Express capability list structure as defined in the *PCI Express* Base Specification*, to indicate the capabilities of the root interconnect.

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 985. Register Summary: Root Port Capability Structure

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
40h	41h	CLIST	Capabilities List Register	8010h	RO
42h	43h	XCAP	PCI Express Capabilities Register	0041h	RO
44h	47h	DCAP	Device Capabilities Register	00000FE0h	RO
48h	49h	DCTL	Device Control Register	0000h	RO, RW
4Ah	4Bh	DSTS	Device Status Register	0010h	RO, RWC
4Ch	4Fh	LCAP	Link Capabilities Register	see desc	RO
50h	51h	LCTL	Link Control Register	0000h	RO, RW
52h	53h	LSTS	Link Status Register	see desc	RO
54h	57h	SLCAP	Slot Capabilities Register	00000060h	RO, RWO
58h	59h	SLCTL	Slot Control Register	0000h	RO, RW
5Ah	5Bh	SLSTS	Slot Status Register	0x000000h	RO, RWC
5Ch	5Dh	RCTL	Root Control Register	0000h	RO, RW
60h	63h	RSTS	Root Status Register	*00000000h*	RO

30.2.2.1 Offset 40 - 41h: CLIST – Capabilities List Register

Table 986. Offset 40 - 41h: CLIST – Capabilities List Register

Device: 28 ***Function:*** 0, 1, 2, 3
Offset: 40 - 41h ***Size:*** *16 bit*
Default Value: *8010h* ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:08	NEXT	**Next Capability:** Value of 80h indicates the location of the next pointer.	80h	RO
07:00	CID	**Capability ID:** Indicates this is a PCI Express capability.	10h	RO

30.2.2.2 Offset 42 - 43h: XCAP – PCI Express Capabilities Register

Table 987. Offset 42 - 43h: XCAP – PCI Express Capabilities Register

Device: 28 ***Function:*** 0, 1, 2, 3
Offset: 42 - 43h ***Size:*** *16 bit*
Default Value: *0041h* ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:14	Reserved	Reserved	0	
13:09	IMN	**Interrupt Message Number:** There are not multiple MSI interrupt numbers.	0	RO

Table 987. Offset 42 - 43h: XCAP – PCI Express Capabilities Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 42 - 43h	***Size:*** *16 bit*
Default Value: *0041h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
08	SI	**Slot Implemented:** Indicates whether the root port is connected to a slot. Slot support is platform specific. BIOS programs this field, and it is maintained until a platform reset.	0	RWO
07:04	DT	**Device / Port Type:** Indicates this is a PCI Express root port.	4h	RO
03:00	CV	**Capability Version:** Indicates PCI Express 1.0	1h	RO

30.2.2.3 Offset 44 - 47h: DCAP – Device Capabilities Register

Table 988. Offset 44 - 47h: DCAP – Device Capabilities Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 44 - 47h	***Size:*** *32 bit*
Default Value: *00000FE0h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:28	Reserved	Reserved	0	
27:26	CSPS	**Captured Slot Power Limit Scale:** Not supported	0	
25:18	CSPV	**Captured Slot Power Limit Value:** Not supported	0	
17:15	Reserved	Reserved	0	
14	PIP	**Power Indicator Present:** Indicates no power indicator is present on the root port.	0	RO
13	AIP	**Attention Indicator Present:** Indicates no attention indicator is present on the root port.	0	RO
12	ABP	**Attention Button Present:** Indicates no attention button is present on the root port.	0	RO
11:09	E1AL	**Endpoint L1 Acceptable Latency:** Indicates more than 4 µs. This field essentially has no meaning for root ports since root ports are not endpoints. The Intel® 3100 Chipset is a root port.	111	RO
08:06	E0AL	**Endpoint L0 Acceptable Latency:** Indicates more than 64 µs. This field essentially has no meaning for root ports since root ports are not endpoints. The Intel® 3100 Chipset is a root port.	111	RO
05	ETFS	**Extended Tag Field Supported:** Indicates that 8-bit tag fields are supported.	1	RO
04:03	PFS	**Phantom Functions Supported:** No phantom functions supported.	00	RO
02:00	MPS	**Maximum Payload Size Supported:** Indicates the maximum payload size supported is 128 B.	000	RO

30.2.2.4 Offset 48 - 49h: DCTL – Device Control Register

Table 989. Offset 48 - 49h: DCTL – Device Control Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 48 - 49h	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15	Reserved	Reserved	0	
14:12	MRRS	**Maximum Read Request Size:** Hardwired to 0.	000	RO
11	ENS	**Enable No Snoop:** Not supported. The root port never issues non-snoop requests.	0	RO
10	APME	**Aux Power PM Enable:** Must be read/write for OS testing. The OS sets this bit to '1' if the device connected has detected aux power. It has no effect on the root port otherwise.	0	RW
09	PFE	**Phantom Functions Enable:** Not supported.	0	
08	ETFE	**Extended Tag Field Enable:** Not supported.	0	
07:05	MPS	**Maximum Payload Size:** The root port only supports 128 B payloads, regardless of the programming of this field.	000	RW
04	ERO	**Enable Relaxed Ordering:** Not supported.	0	
03	URE	**Unsupported Request Reporting Enable:** 0 = The root port ignores unsupported request errors. 1 = The root port generates errors when detecting an unsupported request.	0	RW
02	FEE	**Fatal Error Reporting Enable:** 0 = The root port ignores fatal errors. 1 = The root port generates errors when detecting a fatal error.	0	RW
01	NFE	**Non-Fatal Error Reporting Enable:** 0 = The root port ignores non-fatal errors. 1 = The root port generates errors when detecting a non-fatal error.	0	RW
00	CEE	**Correctable Error Reporting Enable:** 0 = The root port ignores correctable errors. 1 = The root port generates errors when detecting a correctable error.	0	RW

30.2.2.5 Offset 4A - 4Bh: DSTS – Device Status Register

Table 990. Offset 4Ah: DSTS – Device Status Register (Sheet 1 of 2)

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 4A - 4Bh	***Size:*** *16 bit*
Default Value: *0010h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:06	Reserved	Reserved	0	
05	TDP	**Transactions Pending:** This bit has no meaning for the root port since only one transaction may be pending to the IICH, so a read of this bit cannot occur until it has already returned to zero.	0	RO
04	APD	**AUX Power Detected:** The root port contains AUX power for wakeup.	1	RO
03	URD	**Unsupported Request Detected:** Indicates an unsupported request was detected.	0	RWC

Table 990. Offset 4Ah: DSTS – Device Status Register (Sheet 2 of 2)

Device: 28	**Function:** 0, 1, 2, 3
Offset: 4A - 4Bh	**Size:** *16 bit*
Default Value: *0010h*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
02	FED	**Fatal Error Detected:** Indicates a fatal error was detected. 0 = Fatal error has not occurred. 1 = A fatal error occurred from a data link protocol error, link training error, buffer overflow or malformed TLP.	0	RWC
01	NFED	**Non-Fatal Error Detected:** Indicates a non-fatal error was detected. 0 = Non-fatal error has not occurred. 1 = A non-fatal error occurred from a poisoned TLP, unexpected completions, unsupported requests, completer abort, or completer timeout.	0	RWC
00	CED	**Correctable Error Detected:** Indicates a correctable error was detected. 0 = Correctable error has not occurred. 1 = The port received an internal correctable error from receiver errors / framing errors, TLP CRC error, DLLP CRC error, replay num rollover, replay timeout.	0	RWC

30.2.2.6 Offset 4C - 4Fh: LCAP – Link Capabilities Register

Table 991. Offset 4C - 4Fh: LCAP – Link Capabilities Register (Sheet 1 of 2)

Device: 28	**Function:** 0, 1, 2, 3
Offset: 4C - 4Fh	**Size:** *32 bit*
Default Value: *See bit descriptions*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:24	PN	**Port Number:** Indicates the port number for the root port. This value is different for each implemented port: Port # — Value of PN Field 1 — 01h 2 — 02h 3 — 03h 4 — 04h	See Table	RO
23:18	Reserved	Reserved	0	
17:15	EL1	The Intel® 3100 Chipset does not support L1. **L1 Exit Latency:** Set to 010b to indicates an exit latency of 2 µs to 4 µs.	010	RO

Table 991. Offset 4C - 4Fh: LCAP – Link Capabilities Register (Sheet 2 of 2)

Device: 28	**Function:** 0, 1, 2, 3
Offset: 4C - 4Fh	**Size:** *32 bit*
Default Value: *See bit descriptions*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
14:12	EL0	The Intel® 3100 Chipset does not support L0s. **L0s Exit Latency:** Indicates an exit latency based upon common-clock configuration: **LCTL.CCC** — **Value** 0 — MPC.UCEL 1 — MPC.CCEL ***Note:*** LCLT.CCC is at D28, F0/F1, 50h, bit.	See Table	RO
11:10	APMS	The Intel® 3100 Chipset does not support L0s or L1. **Active State Link PM Support:** Indicates the level of active state power management on this link. (Per PCI Express spec, L0s must be supported, but the Intel® 3100 Chipset has defeatured L0s and L1.) **Bits** — **Definition** 00 — Reserved 01 — Reserved 10 — Reserved 11 — Reserved	3h	RWO
09:04	MLW	**Maximum Link Width (MLW):** For the root ports, several values can be taken, based upon the value of the configuration register field RPC.PC.Refer to Section 14.1.4.1, "Offset 0224 - 0227h: RPC – Root Port Configuration Register": **Port #** — **Value of PN field** **RPC.PC** — **00** — **01** — **10** — **11** 1 — 01h — 02h — 02h — 04h 2 — 01h — 01h — 01h — 01h 3 — 01h — 01h — 02h — 01h 4 — 01h — 01h — 01h — 01h	See Table	RO
03:00	MLS	**Maximum Link Speed:** Indicates the link speed is 2.5 Gbytes/s.	1h	RO

30.2.2.7 Offset 50 - 51h: LCTL – Link Control Register

Table 992. Offset 50 - 51h: LCTL – Link Control Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 50 - 51h	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:08	Reserved	Reserved	0	
07	ES	**Extended Synch:** 0 = Extended synch disabled. 1 = Forces extended transmission of FTS ordered sets in FTS and extra TS2 at exit from L1 prior to entering L0.	0	RW
06	CCC	**Common Clock Configuration:** 0 = The IICH and device are not using a common reference clock. 1 = The IICH and device are operating with a distributed common reference clock.	0	RW
05	RL	**Retrain Link:** 0 = This bit always returns 0 when read. 1 = The root port trains its downstream link. ***Note:*** Software uses LSTS.LT and LSTS.LTE (Section 30.2.2.8) to check the status of training.	0	WO
04	LD	**Link Disable:** 0 = Link enabled. 1 = The root port will disable the link.	0	RW
03	RCBC	**Read Completion Boundary Control:** Indicates the read completion boundary is 64 bytes.	0	RO
02	Reserved	Reserved	0	
01:00	APMC	The Intel® 3100 Chipset defeatured L0s, and ASPM should never be turned on. External PCI Express devices should never request entrance into L0s. If it does, undefined behavior will result. External device must set ASPM control register to “disable”, or 00b. **Active State Link PM Control:** Indicates whether the root port must enter L0s or L1 or both. **Bits** — **Definition** 00 — Disabled 01 — Reserved 10 — Reserved 11 — Reserved	0h	RW

30.2.2.8 Offset 52 - 53h: LSTS – Link Status Register

Table 993. Offset 52 - 53h: LSTS – Link Status Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 52 - 53h	***Size:*** *16 bit*
Default Value: *see bit descriptions*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:13	Reserved	Reserved	0	
12	SCC	**Slot Clock Configuration: The** IICH uses the same reference clock as on the platform and does not generate its own clock.	1	RO
11	LT	**Link Training:** 0 = Link training completed. 1 = Link training is occurring.	0	RO
10	LTE	**Link Training Error:** Not supported.	0	
09:04	NLW	**Negotiated Link Width:** For the root ports, this register could take on several values: **Port#** / **Value of PN field** **RPC.PC** **00** **01** **10** **11** 1 01h 02h 02h 04h 2 01h 01h 01h 01h 3 01h 01h 02h 01h 4 01h 01h 01h 01h The contents of this register are undefined if the link has not successfully trained.	See Table	RO
03:00	LS	**Link Speed:** This field indicates the negotiated Link speed of the given PCI Express Link. Link is 2.5 Gbytes/s	1h	RO

30.2.2.9 Offset 54 - 57h: SLCAP – Slot Capabilities Register

Table 994. Offset 54 - 57h: SLCAP – Slot Capabilities Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 54 - 57h	***Size:*** *32 bit*
Default Value: *00000060h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:19	PSN	**Physical Slot Number:** This is a value that is unique to the slot number. BIOS sets this field and it remains set until a platform reset.	0000h	RWO
18:17	Reserved	Reserved	0	
16:15	SLS	**Slot Power Limit Scale:** Specifies the scale used for the slot power limit value. BIOS sets this field and it remains set until a platform reset.	00	RWO
14:07	SLV	**Slot Power Limit Value:** Specifies the upper limit (in conjunction with SLS value), on the upper limit on power supplied by the slot. The two values together indicate the amount of power in watts allowed for the slot. BIOS sets this field, and it remains set until a platform reset.	000h	RWO

Table 994. Offset 54 - 57h: SLCAP – Slot Capabilities Register

Device: 28	**Function:** 0, 1, 2, 3
Offset: 54 - 57h	**Size:** 32 bit
Default Value: 00000060h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
06	HPC	**Hot Plug Capable:** 0 = Hot plug is not supported. 1 = Hot plug is supported.	1	RO
05	HPS	**Hot Plug Surprise:** 0 = Device may not be removed from the slot without prior notification. 1 = Device may be removed from the slot without prior notification.	1	RO
04	PIP	**Power Indicator Present:** 0 = Power indicator LED is not present for this slot. 1 = Power indicator LED is present for this slot.	0	RO
03	AIP	**Attention Indicator Present:** 0 = Attention indicator LED is not present for this slot. 1 = Attention indicator LED is present for this slot.	0	RO
02	MSP	**MRL Sensor Present:** 0 = MRL sensor is not present. 1 = MRL sensor is present.	0	RO
01	PCP	**Power Controller Present:** 0 = Power controller is not supported for this slot. 1 = Power controller is supported for this slot.	0	RO
00	ABP	**Attention Button Present:** 0 = Attention button is not supported for this slot. 1 = Attention button is supported for this slot.	0	RO

30.2.2.10 Offset 58 - 59h: SLCTL – Slot Control Register

Table 995. Offset 58 - 59h: SLCTL – Slot Control Register (Sheet 1 of 2)

Device: 28	**Function:** 0, 1, 2, 3
Offset: 58 - 59h	**Size:** 16 bit
Default Value: 0000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
15:11	Reserved	Reserved	0	
10	PCC	**Power Controller Control:** This bit has no meaning for module-based hot plug.	0	RO
09:08	PIC	**Power Indicator Control:** When read, the current state of the power indicator is returned. When written, the appropriate POWER_INDICATOR_* messages are sent. Defined encodings are: Bits — Definition 00b — Reserved 01b — On 10b — Blink 11b — Off	0	RW

Table 995. Offset 58 - 59h: SLCTL – Slot Control Register (Sheet 2 of 2)

Device: 28	**Function:** 0, 1, 2, 3
Offset: 58 - 59h	**Size:** *16 bit*
Default Value: *0000h*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
07:06	AIC	**Attention Indicator Control:** When read, the current state of the attention indicator is returned. When written, the appropriate ATTENTION_INDICATOR_* messages are sent. Defined encodings are: Bits — Definition 00b — Reserved 01b — On 10b — Blink 11b — Off	0	RW
05	HPE	**Hot Plug Interrupt Enable:** 0 = Hot plug interrupts based on hot plug events is disabled. 1 = Enables generation of a hot plug interrupt on enabled hot plug events.	0	RW
04	CCE	**Command Completed Interrupt Enable:** 0 = Hot plug interrupts based on command completions is disabled. 1 = Enables the generation of a hot plug interrupt when a command is completed by the hot plug controller.	0	RW
03	PDE	**Presence Detect Changed Enable:** 0 = Hot plug interrupts based on presence detect logic changes is disabled. 1 = Enables the generation of a hot plug interrupt or wake message when the presence detect logic changes state.	0	RW
02	MSE	**MRL Sensor Changed Enable:** MSE is not supported.	0	
01	PFE	**Power Fault Detected Enable:** PFE is not supported.	0	
00	ABE	**Attention Button Pressed Enable:** 0 = Hot plug interrupts based on the attention button being pressed is disabled. 1 = Enables the generation of a hot plug interrupt when the attention button is pressed.	0	RW

30.2.2.11 Offset 5A - 5Bh: SLSTS – Slot Status Register

Table 996. Offset 5A - 5Bh: SLSTS – Slot Status Register (Sheet 1 of 2)

Device: 28	**Function:** 0, 1, 2, 3
Offset: 5A - 5Bh	**Size:** *16 bit*
Default Value: *0x0000000*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
15:07	Reserved	Reserved	0	
06	PDS	**Presence Detect State:** If XCAP.SI is set (indicating that this root port spawns a slot), then this bit: 0 = Indicates the slot is empty. 1 = Indicates the slot has a device connected. Otherwise, if XCAP.SI is cleared, this bit is always set (1).	See Desc	RO
05	MS	**MRL Sensor State:** Reserved .	0	

Table 996. Offset 5A - 5Bh: SLSTS – Slot Status Register (Sheet 2 of 2)

Device: 28	**Function:** 0, 1, 2, 3
Offset: 5A - 5Bh	**Size:** *16 bit*
Default Value: *0x0000000*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
04	CC	**Command Completed:** 0 = Issued command not completed. 1 = The hot plug controller completed an issued command. This is set when the last message of a command is sent and indicates that software can write a new command to there slot control register.	0	RWC
03	PDC	**Presence Detect Changed: This bit is set by the root port when the PD bit changes state.** 0 = No change in the PD bit. 1 = The PD bit changed states.	0	RWC
02	MSC	**MRL Sensor Changed:** Reserved.	0	
01	PFD	**Power Fault Detected:** Reserved.	0	
00	ABP	**Attention Button Pressed:** 0 = The attention button has not been pressed. 1 = The attention button is pressed.	0	RWC

30.2.2.12 Offset 5C - 5Dh: RCTL – Root Control Register

Table 997. Offset 5C - 5Dh: RCTL – Root Control Register

Device: 28	**Function:** 0, 1, 2, 3
Offset: 5C - 5Dh	**Size:** *16 bit*
Default Value: *0000h*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
15:04	Reserved	Reserved	0	
03	PIE	**PME Interrupt Enable:** 0 = Interrupt generation disabled. 1 = Interrupt generation enabled when RSTS.PS (Section 30.2.2.13) is in a set state (either due to a 0 to 1 transition, or due to this bit being set with RSTS.PS already set).	0	RW
02	SFE	**System Error on Fatal Error Enable:** 0 = An SERR# will not be generated. 1 = An SERR# will be generated, assuming CMD.SEE (Section 30.2.1.2) is set, if a fatal error is reported by any of the devices in the hierarchy of this root port, including fatal errors in this root port.	0	RW
01	SNE	**System Error on Non-Fatal Error Enable:** 0 = An SERR# will not be generated. 1 = An SERR# will be generated, assuming CMD.SEE (Section 30.2.1.2) is set, if a non-fatal error is reported by any of the devices in the hierarchy of this root port, including non-fatal errors in this root port.	0	RW
00	SCE	**System Error on Correctable Error Enable:** 0 = An SERR# will not be generated. 1 = An SERR# will be generated, assuming CMD.SEE (Section 30.2.1.2) if a correctable error is reported by any of the devices in the hierarchy of this root port, including correctable errors in this root port.	0	RW

30.2.2.13 Offset 60 - 63h: RSTS – Root Status Register

Table 998. Offset 60 - 63h: RSTS – Root Status Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 60 - 63h	***Size:*** *32 bit*
Default Value: *00000000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:18	Reserved	Reserved	0	
17	PP	**PME Pending:** 0 = When the original PME is cleared by software, it will be set again, the requestor ID will be updated, and this bit will be cleared. 1 = Indicates another PME is pending when the PME status bit is set.	0	RO
16	PS	**PME Status:** 0 = PME was not asserted. 1 = Indicates that PME was asserted by the requestor ID in RID. Subsequent PMEs are kept pending until this bit is cleared.	0	RWC
15:00	RID	**PME Requestor ID:** Indicates the PCI requestor ID of the last PME requestor. Valid only when PS is set.	0	RO

30.2.3 Message Signaled Interrupt Capability

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 999. Register Summary: Message Signaled Interrupt Capability

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
80h	81h	MID	Message Signaled Interrupt Identifiers	*9005h*	RO
82h	83h	MC	Message Signaled Interrupt Control	*0000h*	RW, RO
84h	87h	MA	Message Signaled Interrupt Address	*00000000h*	RW, RO
88h	89h	MD	Message Signaled Interrupt Data	00h	RW, RO

30.2.3.1 Offset 80 - 81h: MID – Message Signaled Interrupt Identifiers Register

Table 1000. Offset 80 - 81h: MID – Message Signaled Interrupt Identifiers Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 80 - 81h	***Size:*** *16 bit*
Default Value: *9005h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:08	NEXT	**Next Pointer:** Indicates the location of the next pointer in the list.	90h	RO
07:00	CID	**Capability ID:** Capabilities ID indicates MSI.	05h	RO

30.2.3.2 Offset 82 - 83h: MC – Message Signaled Interrupt Message Control Register

Table 1001. Offset 82 - 83h: MC – Message Signaled Interrupt Message Control Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 82 - 83h	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:08	Reserved	Reserved	0	
07	C64	**64 Bit Address Capable:** Capable of generating a 32-bit message only.	0	RO
06:04	MME	**Multiple Message Enable:** These bits are RW for software compatibility, but only one message is ever sent by the root port.	000	RW
03:01	MMC	**Multiple Message Capable:** Only one message is required.	000	RO
00	MSIE	**MSI Enable:** 0 = MSI is disabled. 1 = MSI is enabled and traditional interrupt pins are not used to generate interrupts. ***Note:*** CMD.BME (Section 30.2.1.2) must be set for an MSI to be generated. If CMD.BME is cleared, and this bit is set, no interrupts (not even pin based) are generated.	0	RW

30.2.3.3 Offset 84 - 87h: MA – Message Signaled Interrupt Message Address Register

Table 1002. Offset 84 - 87h: MA – Message Signaled Interrupt Message Address Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 84 - 87h	***Size:*** *32 bit*
Default Value: *00000000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:02	ADDR	**Address:** Lower 32 bits of the system specified message address, always DW aligned.	00h	RW
01:00	Reserved	Reserved	00b	

30.2.3.4 Offset 88 - 89h: MD – Message Signaled Interrupt Message Data Register

Table 1003. Offset 88 - 89h: MD – Message Signaled Interrupt Message Data Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 88 - 89h	***Size:*** *16 bit*
Default Value: *00h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:00	DATA	**Data:** This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the lower word (PCI AD[15:00]) during the data phase of the MSI memory write transaction.	00h	RW

30.2.4 PCI Bridge Vendor Capability

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 1004. Register Summary: PCI Bridge Vendor Capability

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
90h	91h	SVCAP	Subsystem Vendor Capability Pointer Register	*A00Dh*	RO
94h	97h	SVID	Subsystem Vendor IDs Register	*00000000h*	RWO

30.2.4.1 Offset 90 - 91h: SVCAP – Subsystem Vendor Capability Register

Table 1005. Offset 90 - 91h: SVCAP – Subsystem Vendor Capability Register

Device: 28 ***Function:*** 0, 1, 2, 3
Offset: 90 - 91h ***Size:*** *16 bit*
Default Value: *A00Dh* ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:08	NEXT	**Next Capability:** Indicates the location of the next pointer in the list.	A0h	RO
07:00	CID	**Capability Identifier:** Value of 0Dh indicates this is a PCI bridge subsystem vendor capability.	0Dh	RO

30.2.4.2 Offset 94 - 97h: SVID – Subsystem Vendor IDs Register

Table 1006. Offset 94 - 97h: SVID – Subsystem Vendor IDs Register

Device: 28 ***Function:*** 0, 1, 2, 3
Offset: 94 - 97h ***Size:*** *32 bit*
Default Value: *00000000h* ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:16	SID	**Subsystem Identifier:** Indicates the subsystem as identified by the vendor. This field is write once and is locked down until a bridge reset occurs (not the PCI bus reset).	0000h	RWO
15:00	SVID	**Subsystem Vendor Identifier:** Indicates the manufacturer of the subsystem. This field is write once and is locked down until a bridge reset occurs (not the PCI bus reset).	0000h	RWO

30.2.5 PCI Power Management Capability

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 1007. Register Summary: PCI Power Management Capability

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
A0h	A1h	PMCAP	Power Management Capability Pointer Register	*0001h*	RO
A2h	A3h	PMC	Power Management Capabilities Register	*C802h*	RO
A4h	A7h	PMCS	Power Management Control and Status Register	*00000000h*	RO

30.2.5.1 Offset A0 - A1h: PMCAP – Power Management Capability Register

Table 1008. Offset A0 - A1h: PMCAP – Power Management Capability Register

Device: 28 ***Function:*** 0, 1, 2, 3
Offset: A0 - A1h ***Size:*** *16 bit*
Default Value: *0001h* ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:08	NEXT	**Next Capability:** Indicates this is the last item in the list.	00h	RO
07:00	CID	**Capability Identifier:** Value of 01h indicates this is a PCI power management capability.	01h	RO

30.2.5.2 Offset A2 - A3h: PMC – PCI Power Management Capabilities Register

Table 1009. Offset A2 - A3h: PMC – PCI Power Management Capabilities Register

Device: 28 ***Function:*** 0, 1, 2, 3
Offset: A2 - A3h ***Size:*** *16 bit*
Default Value: *C802h* ***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:11	PMES	**PME_Support:** Indicates PME# is supported for states D0, D3HOT and D3COLD. The root port does not generate PME#, but reporting that it does is necessary for legacy Microsoft* operating systems to enable PME# in devices connected behind this root port.	11001b	RO
10	D2S	**D2_Support:** The D2 state is not supported.	0b	RO
09	D1S	**D1_Support:** The D1 state is not supported.	0b	RO
08:06	AC	**Aux_Current:** Reports 375 mA maximum suspend well current required when in the $D3_{COLD}$ state.	000b	RO
05	DSI	**Device Specific Initialization:** Indicates that no device-specific initialization is required.	0b	RO
04	Reserved	Reserved	0b	
03	PMEC	**PME Clock:** Indicates that PCI clock is not required to generate PME#.	0b	RO
02:00	VS	**Version:** Indicates support for Revision 1.1 of the *PCI Power Management Specification.*	010b	RO

30.2.5.3 Offset A4 - A7h: PMCS – PCI Power Management Control And Status Register

Table 1010. Offset A4 - A7h: PMCS – PCI Power Management Control And Status Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: A4 - A7h	***Size:*** *32 bit*
Default Value: *00000000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:24	DTA	Reserved	0h	
23	BPCE	**Bus Power / Clock Control Enable:** Reserved per the *PCI Express Base Specification*.	0b	
22	B23S	**B2/B3 Support:** Reserved per the *PCI Express Base Specification*.	0b	
21:16	Reserved	Reserved	0h	
15	PMES	**PME Status:** 0 = No PME received on the downstream link. 1 = Indicates a PME was received on the downstream link.	0h	RO
14:13	DSC	**Data Scale:** Reserved	0h	
12:09	DSEL	**Data Select:** Reserved	0h	
08	PMEE	**PME Enable:** Indicates PME is enabled. The root port takes no action on this bit, but it must be RW for legacy Microsoft operating systems to enable PME# on devices connected to this root port. This register resides in the resume well. The reset for this register is RSMRST# which is not asserted during a Warm Reset.	0b	RWS
07:02	Reserved	Reserved	0h	
01:00	PS	**Power State:** This field is used both to determine the current power state of the root port and to set a new power state. The values are: 00 – D0 state 11 – $D3_{HOT}$ state When in the $D3_{HOT}$ state, the controller's configuration space is available, but the I/O and memory spaces are not. Additionally, interrupts are blocked. If software attempts to write a '10' or '01' to these bits, the write will be ignored. Type 1 configuration cycles are also not accepted. Interrupts are not required to be blocked as software will disable interrupts prior to placing the port into $D3_{HOT}$.	00b	RW

30.2.6 Port Configuration

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 1011. Register Summary: Port Configuration Capability

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
D8h	DBh	MPC	Miscellaneous Port Configuration Register	*00110000h*	RO, RW
DCh	DFh	SMSCS	SMI / SCI Status Register	*00000000h*	RO, RWC

30.2.6.1 Offset D8 - DBh: MPC – Miscellaneous Port Configuration Register

Table 1012. Offset D8 - DBh: MPC – Miscellaneous Port Configuration Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: D8 - DBh	***Size:*** *32 bit*
Default Value: *00110000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31	PMCE	Power Management SCI Enable (PMCE) 0 = SCI generation based on a power management event is disabled. 1 = Enables the root port to generate SCI whenever a power management event is detected.	0b	RW
30	HPCE	Hot Plug SCI Enable (HPCE) 0 = SCI generation based on a hot plug event is disabled. 1 = Enables the root port to generate SCI whenever a hot plug event is detected.	0b	RW
29:08	Reserved	Reserved	0h	
07	PAE	**Port I/OxApic Enable:** 0 = The hole is disabled. 1 = A range is opened through the bridge for the following memory addresses: Port # Address 1 FEC1_0000h - FEC1_7FFFh 2 FEC1_8000h - FEC1_FFFFh 3 FEC2_0000h - FEC2_7FFFh 4 FEC2_8000h - FEC2_FFFFh	0b	RW
06:02	Reserved	Reserved	0b	
01	HPME	Hot Plug SMI Enable (HPME) 0 = SMI generation based on a hot plug event is disabled. 1 = Enables the root port to generate SMI whenever a hot plug event is detected.	0b	RW
00	PMME	Power Management SMI Enable (PMME) 0 = SMI generation based on a power management event is disabled. 1 = Enables the root port to generate SMI whenever a power management event is detected.	0b	RW

30.2.6.2 Offset DC - DFh: SMSCS – SMI / SCI Status Register

Table 1013. Offset DC - DFh: SMSCS – SMI / SCI Status Register

Device: 28	**Function:** 0, 1, 2, 3
Offset: DC - DFh	**Size:** *32 bit*
Default Value: *00000000h*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31	PMCS	**Power Management SCI Status:** 0 = Hot Plug Controller interrupt generation disabled during Hot Plug event. 1 = Hot Plug controller will generate an interrupt during a Hot Plug event. This interrupt has been routed to generate an SCI. See Chapter 30.3.4.4	0b	RWC
30	HPCS	**Hot Plug SCI Status:** 0 = Hot Plug Controller interrupt generation disabled during Hot Plug event. 1 = Hot Plug controller will generate an interrupt during a Hot Plug event. This interrupt has been routed to generate an SCI. See Chapter 30.3.4.4	0b	RWC
29:04	Reserved	Reserved	0h	
03	HPCCM	**Hot Plug Command Completed SMI Status:** 0 = No SLSTS.CC (D28:F0/F1:5A, bit 4) transition detected. 1 = When SLSTS.CC transitions from ‘0’ to ‘1’, and MPC.HPME (D28:F0/F1:D8, bit 1) is ‘1’. In addition, an SMI# will be generated. See Section 30.3.4.4 for more details.	0b	RWC
02	HPABM	**Hot Plug Attention Button SMI Status:** 0 = No SLSTS.ABP (D28:F0/F1:5A, bit 0) transition detected. 1 = When SLSTS.ABP transitions from ‘0’ to ‘1’, and MPC.HPME (D28:F0/F1:D8, bit 1) is ‘1’. In addition an SMI# will be generated. See Section 30.3.4.4 for more details.	0b	RWC
01	HPPDM	**Hot Plug Presence Detect SMI Status:** 0 = No SLSTS.PDC (D28:F0/F1:5A, bit 3) transition detected. 1 = This bit is set when SLSTS.PDC transitions from ‘0’ to ‘1’, and MPC.HPME (D28:F0/F1:D8, bit 1) is set. When this bit is set, an SMI# will be generated. See Section 30.3.4.4 for more details.	0b	RWC
00	PMMS	**Power Management SMI Status:** 0 = No RSTS.PS (D28:F0/F1:60, bit 16) transition detected. 1 = When RSTS.PS transitions from ‘0’ to ‘1’, and MPC.PMME (D28:F0/F1:D8, bit 1) is ‘1’. See Section 30.3.4.4 for more details.	0b	RWC

30.2.7 Manufacturing Information

30.2.7.1 Offset F8 - FBh: MANID – Manufacturer's ID Register

Table 1014. Offset F8 - FBh: MANID – Manufacturer's ID Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: F8 - FBh	***Size:*** *32 bit*
Default Value: *00010F80h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:24	Reserved	Reserved	00h	
23:16	SID	**Stepping Identifier:** This field increments for each stepping of the part. This field can be used by software to differentiate steppings when the Revision ID may not change. A single Stepping ID can be implemented that is readable from all functions in the chip because all of them increment in lock-step.	01h - A1	RO
15:08	MID	**Manufacturing Identifier:** 0Fh = Intel	0Fh	RO
07:00	Reserved	Reserved.	80h	

30.2.8 VC Configuration

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 1015. Register Summary: VC Configuration Capability

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
100h	103h	VCH	Virtual Channel Capability Header Register	*18010002*	RO
104h	107h	VCAP1	Virtual Channel Capability 1 Register	00000001h	RO
108h	10Bh	VCAP2	Virtual Channel Capability 2 Register	*00000001h*	RO
10Ch	10Dh	PVC	Port VC Control Register	*0000h*	RO, RW
10Eh	10Fh	PVS	Port VC Status Register	*0000h*	RO
110h	113h	V0CAP	VC 0 Resource Capability Register	*00000001h*	RO
114h	117h	V0CTL	VC 0 Resource Control Register	*800000FFh*	RO, RW
11Ah	11Bh	V0STS	VC 0 Resource Status Register	*0000h*	RO

30.2.8.1 Offset 100 - 103h: VCH – Virtual Channel Capability Header Register

Table 1016. Offset 100 - 103h: VCH – Virtual Channel Capability Header Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 100 - 103h	***Size:*** *32 bit*
Default Value: *18010002*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:20	NCO	**Next Capability Offset:** Indicates the next item in the list.	180h	RO
19:16	CV	**Capability Version:** Indicates this is version 1 of the capability structure by the PCI SIG.	1h	RO
15:00	CID	**Capability ID:** Indicates this is the Virtual Channel capability item.	0002h	RO

30.2.8.2 Offset 104 - 107h: VCAP1 – Virtual Channel Capability 1 Register

Table 1017. Offset 104 - 107h: VCAP1 – Virtual Channel Capability 1 Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 104 - 107h	***Size:*** *32 bit*
Default Value: 00000001h	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:12	Reserved	Reserved	0h	
11:10	PATS	**Port Arbitration Table Entry Size:** Must be set to 00 for root ports are per the *PCI Express Base Specification*.	00b	RO
09:08	RC	**Reference Clock:** Fixed at 100 ns for this version of the *PCI Express Base Specification*.	00b	RO
07	Reserved	Reserved	0b	
06:04	LPEVC	**Low Priority Extended VC Count:** Indicates that there are no additional VCs of low priority with extended capabilities	000b	RO
03:00	Reserved	Reserved	0b	

30.2.8.3 Offset 108 - 10Bh: VCAP2 – Virtual Channel Capability 2 Register

Table 1018. Offset 108 - 10Bh: VCAP2 – Virtual Channel Capability 2 Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 108 - 10Bh	***Size:*** *32 bit*
Default Value: *00000001h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:24	ATO	**VC Arbitration Table Offset:** Indicates that no table is present for VC arbitration since it is fixed.	00h	RO
23:08	Reserved	Reserved	0	
07:00	AC	**VC Arbitration Capability:** Indicates that the VC arbitration is fixed in the root port. VC0 is highestpriority.	01h	RO

30.2.8.4 Offset 10C - 10Dh: PVC – Port Virtual Channel Control Register

Table 1019. Offset 10C - 10Dh: PVC – Port Virtual Channel Control Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 10C - 10Dh	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:04	Reserved	Reserved	0h	
03:01	AS	**VC Arbitration Select:** Indicates which VC must be programmed in the VC arbitration table. The root port takes no action on the setting of this field since there is no arbitration table.	000b	RW
00	LAT	**Load VC Arbitration Table:** Indicates that the table programmed must be loaded into the VC arbitration table. This bit is defined as read/write with always returning 0 on reads. Since there is no VC arbitration table in the root port, this bit can be built as RO.	0b	RO

30.2.8.5 Offset 10E - 10Fh: PVS – Port Virtual Channel Status Register

Table 1020. Offset 10E - 10Fh: PVS – Port Virtual Channel Status Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 10E - 10Fh	***Size:*** *16 bit*
Default Value: *0000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
15:01	Reserved	Reserved	0h	
00	VAS	**VC Arbitration Table Status:** Indicates the coherency status of the VC Arbitration table when it is being updated. This field is always 0 in the root port since there is no VC arbitration table.	0b	RO

30.2.8.6 Offset 110 - 113h: V0CAP – Virtual Channel 0 Resource Capability Register

Table 1021. Offset 110 - 113h: V0CAP – Virtual Channel 0 Resource Capability Register (Sheet 1 of 2)

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 110 - 113h	***Size:*** *32 bit*
Default Value: *00000001h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:24	AT	**Port Arbitration Table Offset:** This VC implements no port arbitration table since the arbitration is fixed.	0	RO
23	Reserved	Reserved	0	
22:16	MTS	**Maximum Time Slots:** This VC implements fixed arbitration, and therefore this field is not used.	00h	RO

Table 1021. Offset 110 - 113h: V0CAP – Virtual Channel 0 Resource Capability Register (Sheet 2 of 2)

Device: 28	**Function:** 0, 1, 2, 3
Offset: 110 - 113h	**Size:** 32 bit
Default Value: 00000001h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
15	RTS	**Reject Snoop Transactions:** This VC must be able to take snoopable transactions. 0 = Reject Snoop 1 = Accept Snoop	0	RO
14	APS	**Advanced Packet Switching:** This VC is capable of all transactions, not just advanced packet switching transactions. 0 = Capable of all transactions 1 = Capable of only advanced packet switching transactions	0	RO
13:08	Reserved	Reserved	0	
07:00	PAC	**Port Arbitration Capability:** 00= Reserved01= VC uses fixed port arbitration. 10= Reserved 11= Reserved	01h	RO

30.2.8.7 Offset 114 - 117h: V0CTL – Virtual Channel 0 Resource Control Register

Table 1022. Offset 114 - 117h: V0CTL – Virtual Channel 0 Resource Control Register (Sheet 1 of 2)

Device: 28	**Function:** 0, 1, 2, 3
Offset: 114 - 117h	**Size:** 32 bit
Default Value: 800000FFh	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31	EN	**Virtual Channel Enable:** 0 = Disables the VC. 1 = Enables the VC. Virtual Channel 0 cannot be disabled.	1	RO
30:27	Reserved	Reserved	0	
26:24	VCID	**Virtual Channel Identifier:** Indicates the ID to use for this virtual channel.	000	RO
23:20	Reserved	Reserved	0	
19:17	PAS	**Port Arbitration Select:** Indicates which port table is being programmed. The root complex takes no action on this setting since the arbitration is fixed and there is no arbitration table.	0	RW

Table 1022. Offset 114 - 117h: V0CTL – Virtual Channel 0 Resource Control Register (Sheet 2 of 2)

Device: 28	**Function:** 0, 1, 2, 3
Offset: 114 - 117h	**Size:** *32 bit*
Default Value: *800000FFh*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
16	LAT	**Load Port Arbitration Table:** The root port does not implement an arbitration table for this virtual channel.	0	
15:08	Reserved	Reserved	0	
07:00	TCVC0M	**Transaction Class / Virtual Channel 0 Map:** Indicates which transaction classes are mapped to the virtual channel 0. When a bit is '1', this transaction class is mapped to the virtual channel 0. Bit Transaction Class 7 Transaction Class 7 6 Transaction Class 6 5 Transaction Class 5 4 Transaction Class 4 3 Transaction Class 3 2 Transaction Class 2 1 Transaction Class 1 ***Notes:*** 1. Transaction Class 0 is reserved and must always be mapped to VC0 (Bit 0 = 1). 2. Bit 0 access is RO. Bits 7:1 are RW.	7Fh	RW/RO (See Note 2)

30.2.8.8 Offset 11A - 11Bh: V0STS – Virtual Channel 0 Resource Status Register

Table 1023. Offset 11A - 11Bh: V0STS – Virtual Channel 0 Resource Status Register

Device: 28	**Function:** 0, 1, 2, 3
Offset: 11A - 11Bh	**Size:** *16 bit*
Default Value: *0000h*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
15:02	Reserved	Reserved	0	
01	NP	**VC Negotiation Pending:** 0 = Negotiation is not pending. 1 = Indicates the virtual channel is still being negotiated with ingress ports.	0	RO
00	ATS	**Port Arbitration Tables Status:** There is no port arbitration table for this VC, so this bit is reserved at 0.	0	

30.2.9 Advanced Error Reporting Configuration

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 1024. Register Summary: Advanced Error Reporting Capability

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
140h	143h	AECH	Advanced Error Reporting Capability Header Register	*0000_0000h*	RO
144h	147h	UES	Uncorrectable Error Status Register	*0000_0000h*	RO, RWC
148h	14Bh	UEM	Uncorrectable Error Mask Register	*0000_0000h*	RO, RWO
14Ch	14Dh	UEV	Uncorrectable Error Severity Register	*0006_0011h*	RO
150h	153h	CES	Correctable Error Status Register	0000_0000h	RO, RWC
154h	157h	CEM	Correctable Error Mask Register	0000_0000h	RO, RWC
158h	15Bh	AECC	Advanced Error Capabilities and Control Register	0000_0000h	RO
15Ch	16Bh	HL	Header Log Register	0000_0000h	RO
16Ch	16Fh	REC	Root Error Command Register	0000_0000h	RO
170h	173h	RES	Root Error Status Register	0000_0000h	RO

30.2.9.1 Offset 140 - 143h: AECH – Advanced Error Reporting Capability Header Register

For an exposed capability, the capability ID would be 1h to indicate that this is an AER capability. For this implementation, since AER is not exposed, the register is reserved.

Table 1025. Offset 140 - 143h: AECH – Advanced Error Reporting Capability Header Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 140 - 143h	***Size:*** *32 bit*
Default Value: *0000_0000*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:20	NCO	**Next Capability Offset:** Reserved.	000h	
19:16	CV	**Capability Version:** Reserved.	0h	
15:00	CID	**Capability ID:** Reserved.	0000h	

30.2.9.2 Offset 144 - 147h: UES – Uncorrectable Error Status Register

This register must maintain its state through a platform reset. It loses its state upon suspend.

Table 1026. Offset 144 - 147h: UES – Uncorrectable Error Status Register

Device: 28	**Function:** 0, 1, 2, 3
Offset: 144 - 147h	**Size:** 32 bit
Default Value: 0000_0000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:21	Reserved	Reserved	000h	
20	URE	**Unsupported Request Error Status: Indicates an unsupported request was received.**	0b	
19	EE	**ECRC Error Status:** ECRC is not supported.	0b	
18	MT	**Malformed TLP Status:** Indicates a malformed TLP was received.	0b	RWC
17	RO	**Receiver Overflow Status:** Indicates a receiver overflow occurred.	0b	RWC
16	UC	**Unexpected Completion Status:** Indicates an unexpected completion was received.	0b	RWC
15	CA	**Completer Abort Status:** Indicates a completer abort was received.	0b	RWC
14	CT	**Completion Timeout Status:** Indicates a completion timed out.	0b	RWC
13	FCPE	**Flow Control Protocol Error Status:** Not supported.	0b	
12	PT	**Poisoned TLP Status:** Indicates a poisoned TLP was received.	0b	RWC
11:05	Reserved	Reserved	00h	
04	DLPE	**Data Link Protocol Error Status:** Indicates a data link protocol error occurred.	0b	
03:01	Reserved	Reserved	0h	
00	TE	**Training Error Status:** Not supported.	0b	

30.2.9.3 Offset 148 -14Bh: UEM – Uncorrectable Error Mask Register

When set ('1'), the corresponding error in the UES register is masked, and the logged error will cause no action. When cleared ('0'), the corresponding error is enabled.

Table 1027. Offset 148 -14Bh: UEM – Uncorrectable Error Mask Register (Sheet 1 of 2)

Device: 28	**Function:** 0, 1, 2, 3
Offset: 148 - 14Bh	**Size:** 32 bit
Default Value: 0000_0000	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:21	Reserved	Reserved	0h	
20	URE	**Unsupported Request Error Mask:** Mask for uncorrectable errors. 0 = The corresponding error in the UES register (D28:F0/F1:144) is enabled. 1 = The corresponding error in the UES register (D28:F0/F1:144) is masked.	0b	RWO
19	EE	**ECRC Error Mask:** ECRC is not supported.	0b	RO
18	MT	**Malformed TLP Mask:** Mask for malformed TLPs. 0 = The corresponding error in the UES register (D28:F0/F1:144) is enabled. 1 = The corresponding error in the UES register (D28:F0/F1:144) is masked.	0b	RWO

Table 1027. Offset 148 -14Bh: UEM – Uncorrectable Error Mask Register (Sheet 2 of 2)

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 148 - 14Bh	***Size:*** *32 bit*
Default Value: *0000_0000*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
17	RO	**Receiver Overflow Mask:** Mask for receiver overflows. 0 = The corresponding error in the UES register (D28:F0/F1:144) is enabled. 1 = The corresponding error in the UES register (D28:F0/F1:144) is masked.	0b	RWO
16	UC	**Unexpected Completion Mask:** Mask for unexpected completions. 0 = The corresponding error in the UES register (D28:F0/F1:144) is enabled. 1 = The corresponding error in the UES register (D28:F0/F1:144) is masked.	0b	RWO
15	CM	**Completer Abort Severity:** Mask for abort completions. 0 = The corresponding error in the UES register (D28:F0/F1:144) is enabled. 1 = The corresponding error in the UES register (D28:F0/F1:144) is masked.	0b	RWO
14	CT	**Completion Timeout Severity: Mask** for timeout completions. 0 = The corresponding error in the UES register (D28:F0/F1:144) is enabled. 1 = The corresponding error in the UES register (D28:F0/F1:144) is masked.	0b	RWO
13	FCPE	**Flow Control Protocol Error Mask:** Not supported.	0b	RO
12	PT	**Poisoned TLP Mask:** Mask for poisoned TLPs. 0 = The corresponding error in the UES register (D28:F0/F1:144) is enabled. 1 = The corresponding error in the UES register (D28:F0/F1:144) is masked.	0b	RWO
11:05	Reserved	Reserved	0h	
04	DLPE	**Data Link Protocol Error Mask:** Mask for data link protocol errors. 0 = The corresponding error in the UES register (D28:F0/F1:144) is enabled. 1 = The corresponding error in the UES register (D28:F0/F1:144) is masked.	0b	RWO
03:01	Reserved	Reserved	0h	
00	TE	**Training Error Mask:** Not supported.	0b	

30.2.9.4 Offset 14C - 14Fh: UEV – Uncorrectable Error Severity Register

In an exposed AER capability, this register gives the option to make an uncorrectable error fatal or non-fatal. In this implementation, this register is read-only with the severity status set to the default of the *PCI Express Base Specification*. An error is fatal if the bit is set. An error is non-fatal if the bit is cleared.

Table 1028. Offset 14C - 14Fh: UEV – Uncorrectable Error Severity Register

Device: 28	**Function:** 0, 1, 2, 3
Offset: 14C - 14Fh	**Size:** *32 bit*
Default Value: *0006_0011h*	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:21	Reserved	Reserved	0	
20	URE	**Unsupported Request Error Severity:** Severity for unsupported request reception. 0 = Error considered non-fatal. (Default) 1 = Error is fatal.	0	RO
19	EE	**ECRC Error Severity:** ECRC is not supported.	0	RO
18	MT	**Malformed TLP Severity:** Severity for malformed TLP reception. 0 = Error considered non-fatal. 1 = Error is fatal. (Default)	1	RO
17	RO	**Receiver Overflow Severity:** Severity for receiver overflow occurrences. 0 = Error considered non-fatal. 1 = Error is fatal. (Default).	1	RO
16	UC	**Unexpected Completion Severity:** Severity for unexpected completion reception. 0 = Error considered non-fatal. (Default) 1 = Error is fatal.	0	RO
15	CA	**Completer Abort Severity:** Severity for completer abort. 0 = Error considered non-fatal. (Default) 1 = Error is fatal.	0	RO
14	CT	**Completion Timeout Severity:** Severity for completion timeout. 0 = Error considered non-fatal. (Default) 1 = Error is fatal.	0	RO
13	FCPE	**Flow Control Protocol Error Severity:** Not supported.	0	RO
12	PT	**Poisoned TLP Severity:** Severity for poisoned TLP reception. 0 = Error considered non-fatal. (Default) 1 = Error is fatal.	0	RO
11:05	Reserved	Reserved	0	
04	DLPE	**Data Link Protocol Error Severity:** Severity for data link protocol errors. 0 = Error considered non-fatal. 1 = Error is fatal. (Default)	1	RO
03:01	Reserved	Reserved	0	
00	TE	**Training Error Severity:** TE not supported.	1	RO

30.2.9.5 Offset 150 - 153h: CES – Correctable Error Status Register

Table 1029. Offset 150 - 153h: CES – Correctable Error Status Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 150 - 153h	***Size:*** *32 bit*
Default Value: *0000_0000*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:13	Reserved	Reserved	0h	
12	RTT	**Replay Timer Timeout Status:** 0 = Replay timer has not timed out). 1 = Indicates the replay timer timed out.	0b	RWC
11:09	Reserved	Reserved	0h	
08	RNR	**Replay Number Rollover Status:**. 0 = Replay number has not rolled over 1 = Indicates the replay number rolled over	0b	RWC
07	BD	**Bad DLLP Status:** 0 = No bad DLLPs received. 1 = Indicates a bad DLLP was received.	0b	RWC
06	BT	**Bad TLP Status:** 0 = No bad TLPs received. 1 = Indicates a bad TLP was received.	0b	RWC
05:01	Reserved	Reserved	0h	
00	RE	**Receiver Error Status:** 0 = No receiver error. 1 = Indicates a receiver error occurred.	0b	RWC

30.2.9.6 Offset 154 - 157h: CEM – Correctable Error Mask Register

When set ('1'), the corresponding error in the CES register is masked, and the logged error will cause no action. When cleared ('0'), the corresponding error is enabled.

Table 1030. Offset 154 - 157h: CEM – Correctable Error Mask Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 154 - 157h	***Size:*** *32 bit*
Default Value: *0000_0000*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:13	Reserved	Reserved	0h	
12	RTT	**Replay Timer Timeout Mask:** Mask for replay timer timeout.	0b	RWO
11:09	Reserved	Reserved	0h	
08	RNR	**Replay Number Rollover Mask:** Mask for replay number rollover.	0b	RWO
07	BD	**Bad DLLP Mask:** Mask for bad DLLP reception.	0b	RWO
06	BT	**Bad TLP Mask:** Mask for bad TLP reception.	0b	RWO
05:01	Reserved	Reserved	0h	
00	RE	**Receiver Error Mask:** Mask for receiver errors.	0b	RWO

30.2.9.7 Offset 158 - 15Bh: AECC – Advanced Error Capabilities and Control Register

Table 1031. Offset 158 - 15Bh: AECC – Advanced Error Capabilities and Control Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 158 - 15Bh	***Size:*** *32 bit*
Default Value: *0000_0000*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:09	Reserved	Reserved	0h	
08	ECE	**ECRC Check Enable:** ECRC is not supported.	0b	RO
07	ECC	**ECRC Check Capable:** ECRC is not supported.	0b	RO
06	EGE	**ECRC Generation Enable:** ECRC is not supported.	0b	RO
05	EGC	**ECRC Generation Capable:** ECRC is not supported.	0b	RO
04:00	FEP	**First Error Pointer**	0h	RO

30.2.9.8 Offset 15C - 16Bh: HL – Header Log Register

In an exposed AER capability, these registers report the header of an error. In this implementation, these registers are reserved.

Table 1032. Offset 15C - 16Bh: HL – Header Log Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 15C - 16Bh	***Size:*** *128 bit*
Default Value: *0000000000000000 0000000000000000h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
127:96	DW4	**Fourth Dword of TLP:** Reserved	0h	
95:64	DW3	**Third Dword of TLP:** Reserved	0h	
63:32	DW2	**Second Dword of TLP:** Reserved	0h	
31:00	DW1	**First Dword of TLP:** Reserved	0h	

30.2.9.9 Offset 16C - 16Fh: REC – Root Error Command Register

In an exposed AER capability, this register allows errors to generate interrupts. For this implementation this register is reserved.

Table 1033. Offset 16C - 16Fh: REC – Root Error Command Register

Device: 28	**Function:** 0, 1, 2, 3
Offset: 16C - 16Fh	**Size:** 32 bit
Default Value: 0000_0000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:03	Reserved	Reserved	0h	
02	FERE	**Fatal Error Reporting Enable:** 0 = Disabled 1 = The root port will generate an interrupt when a fatal error is reported by the attached device.	0b	RO
01	NERE	**Non-fatal Error Reporting Enable:** 0 = Disabled 1 = The root port will generate an interrupt when a non-fatal error is reported by the attached device.	0b	RO
00	CERE	**Correctable Error Reporting Enable:** 0 = Disabled 1 = The root port will generate an interrupt when a correctable error is reported by the attached device.	0b	RO

30.2.9.10 Offset 170 -173h: RES – Root Error Status Register

In an exposed AER capability, this register can track more than one error and set the "multiple" bits if a second or subsequent error occurs and the first has not been serviced. For this implementation, only one error will be captured.

Table 1034. Offset 170 -173h: RES – Root Error Status Register

Device: 28	**Function:** 0, 1, 2, 3
Offset: 170 - 173h	**Size:** 32 bit
Default Value: 0000_0000h	**Power Well:** Core

Bits	Name	Description	Reset Value	Access
31:27	AEMN	**Advanced Error Interrupt Message Number:** There is only one error interrupt allocated.	0h	RO
26:04	Reserved	Reserved	0h	
03	MENR	**Multiple ERR_FATAL/NONFATAL Received:** 0 = No error message received. 1 = Either a fatal or a non-fatal error is received and the ENR bit is already set.	0b	RO
02	ENR	**ERR_FATAL/NONFATAL Received:** 0 = No error message received. 1 = Either a fatal or a non-fatal error message is received.	0b	RWC
01	MCR	**Multiple ERR_COR Received:** 0 = No error message received. 1 = A correctable error message is received and the CR bit is already set. Only one error will be captured.	0b	RO
00	CR	**ERR_COR Received:** 0 = No error message received. 1 = A correctable error message is received.	0b	RWC

30.2.10 Root Complex Topology Capability Structure

The following registers follow the PCI Express capability list structure as defined in the *PCI Express Base Specification*, to indicate the capabilities of NSI.

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 1035. Register Summary: Root Complex Topology Capability Structure Registers

Offset		Symbol	Register Name/Function	Default	Access
Start	End				
180h	183h	RCTCL	Root Complex Topology Capability List Register	*000010005h*	RO
184h	187h	ESD	Element Self Description Register	see bit desc	RO
190h	193h	ULD	Upstream Link Descriptor Register	*00000001h*	RO
198h	19Fh	ULBA	Upstream Link Base Address Register	see bit desc	RO

30.2.10.1 Offset 180 - 183h: RCTCL – Root Complex Topology Capabilities List Register

Table 1036. Offset 180 - 183h: RCTCL – Root Complex Topology Capabilities List Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 180 - 183h	***Size:*** *32 bit*
Default Value: *000010005h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:20	NEXT	**Next Capability:** Indicates the next item in the list; in this case, end of list.	000h	RO
19:16	CV	**Capability Version:** Indicates the version of the capability structure.	1h	RO
15:00	CID	**Capability ID:** Indicates this is a root complex topology capability.	0005h	RO

30.2.10.2 Offset 184 - 187h: ESD – Element Self Description Register

Table 1037. Offset 184 - 187h: ESD – Element Self Description Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 184 - 187h	***Size:*** *32 bit*
Default Value: *see bit descriptions*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:24	PN	**Port Number:** Indicate the ingress port number for the root port. There is a different value per port: Port # Value 1 01h 2 02h 3 03h 4 04h	See Desc	RO
23:16	CID	**Component ID:** This field returns the value of the ESD.CID field of the chip configuration section that is programmed by platform BIOS, since the root port is in the same component as the RCRB.	See Description	RO
15:08	NLE	**Number of Link Entries:** Indicates one link entry (corresponding to the RCRB).	01h	RO
07:04	Reserved	Reserved	0	
03:00	ET	**Element Type:** Indicates that the element type is a root port.	0h	RO

30.2.10.3 Offset 190 - 193h: ULD – Upstream Link Description Register

Table 1038. Offset 190 - 193h: ULD – Upstream Link Description Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 190 - 193h	***Size:*** *32 bit*
Default Value: *00000001h*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
31:24	PN	**Target Port Number:** Indicates the port number of the RCRB.	00h	RO
23:16	TCID	**Target Component ID:** This field returns the value of the ESD.CID field of the chip configuration section that is programmed by platform BIOS, since the root port is in the same component as the RCRB.	00h	RO
15:02	Reserved	Reserved	0	
01	LT	**Link Type:** Indicates that the link points to the IICH RCRB.	0	RO
00	LV	**Link Valid:** Indicates that this link entry is valid.	1	RO

30.2.10.4 Offset 198 - 19Fh: ULBA – Upstream Link Base Address Register

Table 1039. Offset 198 - 19Fh: ULBA – Upstream Link Base Address Register

Device: 28	***Function:*** 0, 1, 2, 3
Offset: 198 - 19Fh	***Size:*** *64 bit*
Default Value: *see bit descriptions*	***Power Well:*** Core

Bits	Name	Description	Reset Value	Access
63:32	BAU	**Base Address Upper:** The RCRB of the IICH is located in 32-bit space.	0h	RO
31:00	BAL	**Base Address Lower:** This field matches the RCBA register (D31:F0:Offset F0h) value in the PCI-LPC bridge.	See Description	RO

30.3 PCI Express* Operation

30.3.1 Interrupt Generation

The root port generates interrupts on behalf of hot plug and power management events, when enabled. These interrupts can be either pin based or MSIs, when enabled. See Section 30.3.2.3 and Section 30.3.4 to see when the root port generates interrupts.

When an interrupt is generated via the legacy pin, the pin is internally routed to the IICH interrupt controllers. The pin that is driven is based upon the setting of the configuration registers. Specifically, the configuration registers used are the D28IP and D28IR registers.

The following table summarizes interrupt behavior for MSI and wire modes. In the table, “bits” refers to the hot plug and PME interrupt bits.

Table 1040. MSI vs. PCI IRQ Actions

Interrupt Register	Wire-Mode Action	MSI Action
All bits 0	Wire inactive	No action
One or more bits set to 1	Wire active	Send message
One or more bits set to 1, new bit gets set to 1	Wire active	Send message
One or more bits set to 1, software clears some (but not all) bits	Wire active	Send message
One or more bits set to 1, software clears all bits	Wire inactive	No action
Software clears one or more bits, and one or more bits are set on the same clock.	Wire active	Send message

30.3.2 Power Management

30.3.2.1 S3/S5 Support

Software initiates the transition to S3/S5 by performing an I/O write to the Power Management Control register in the IICH. After the I/O write completion has been returned to the processor, the Power Management Controller will signal each root port to send a PME_Turn_Off message on it's downstream link. The device attached to the link will eventually respond with a PME_TO_Ack followed by sending a PM_Enter_L23 DLLP request to enter L23. The IICH root ports and Power Management Controller take no action upon receiving a PME_TO_Ack. When all the IICH and IMCH root ports links

are in state L23, the Intel® 3100 ChipsetPower Management Controller will proceed with the entry into S3/S5. See Chapter 22.0, "Power Management," for additional details for the S3/S5 entry sequence.

Prior to entering S3, software is required to put each device into D3hot. When a device is put into D3hot it will initiate entry into a L1 link state by sending a PM_Enter_L1 DLLP. Thus, under normal operating conditions when the root ports sends the PME_Turn_Off message, the link will be in state L1. However, when the root port is instructed to send the PME_Turn_Off message, it will send it whether or not the link was in L1. Endpoints attached to IICH can make no assumptions about the state of the link prior to receiving a PME_Turn_Off message.

30.3.2.2 Resuming from Suspended State

A PCI Express endpoint can cause a system to resume from a suspended state by asserting the WAKE# signal which is monitored by the IICH power management controller. Beacon is not a supported mechanism to exit from suspend.

30.3.2.3 Device Initiated PM_PME Message

When the system has returned to a working state from a previous low power state, a device requesting service will send a PM_PME message continuously until acknowledge, by the root port. The root port will take different actions depending upon whether this is the first PM_PME has been received, or whether a previous message has been received but not yet serviced by the Operating System.

If this is the first message received (RSTS.PS is cleared, '0'), the root port will set RSTS.PS, and log the PME Requester ID into RSTS.RID. If an interrupt is enabled via RCTL.PIE, an interrupt will be generated. This interrupt can be either a pin or an MSI if MSI is enabled via MC.MSIE. See Section 30.3.2.4 for SMI/SCI generation.

If this is a subsequent message received (RSTS.PS is already set, '1'), the root port will set RSTS.PP and log the PME Requester ID from the message in a hidden register. No other action will be taken.

When the first PME event is cleared by software writing '0' to RSTS.PS, the root port will set RSTS.PS, clear RSTS.PP, and move the requester ID from the hidden register into RSTS.RID.

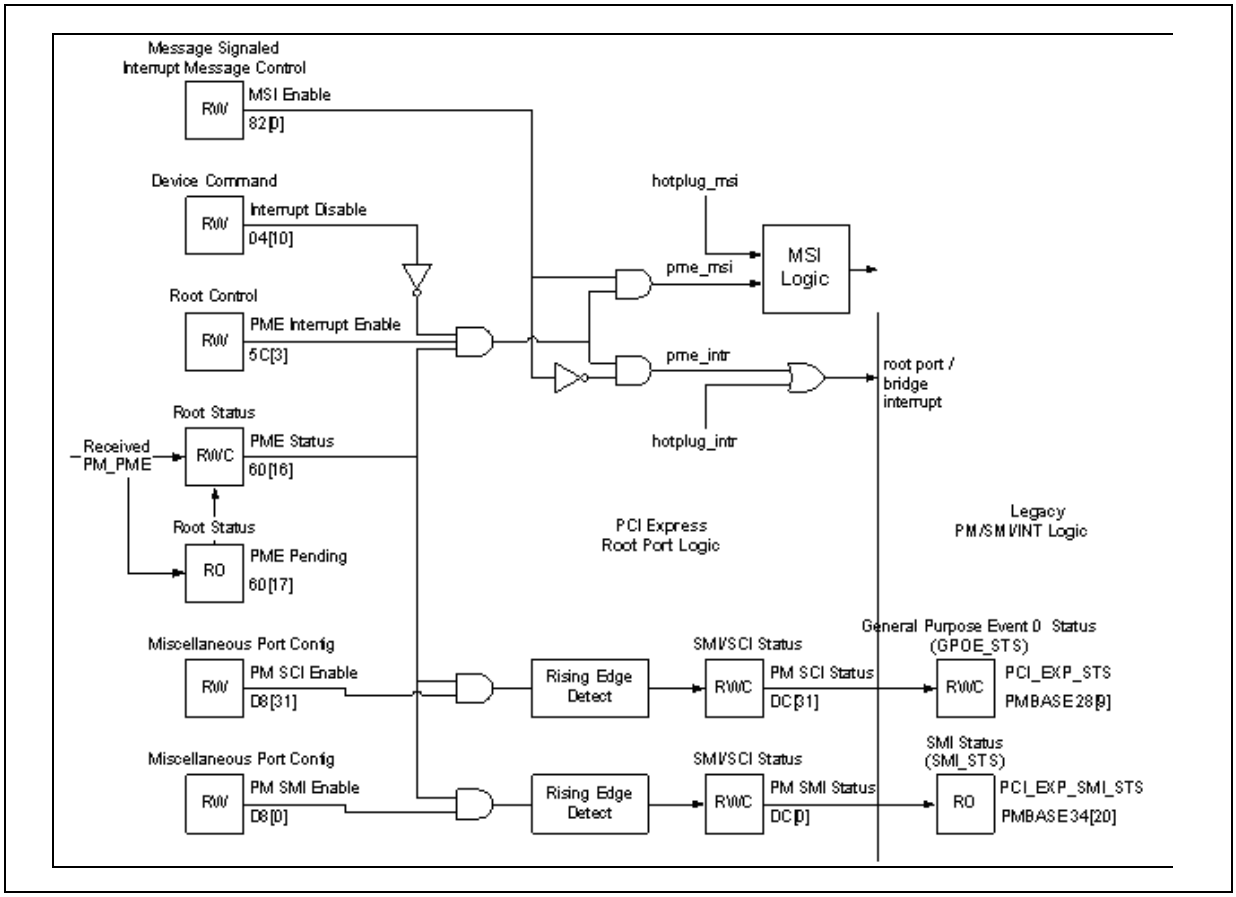
If RCTL.PIE is set, generate an interrupt. If RCTL.PIE is not set, send over to the power management controller so that a GPE can be set. If messages have been logged (RSTS.PS is set), and RCTL.PIE is later written from a '0' to a '1', an interrupt must be generated. This last condition handles the case where the message was received prior to the OS reenabling interrupts after resuming from a low-power state.

30.3.2.4 Power Management SMI/SCI Generation

Interrupts for power management events are not supported on legacy operating systems. To support power management on non PCI Express aware operating systems, PM events can be routed to generate SCI. To generate SCI, MPC.PMCE must be set. When set, a power management event will cause SMSCS.PMCS to be set.

Additionally, BIOS workarounds for power management can be supported by setting MPC.PMME. When this bit is set, power management events will set SMSCS.PMMS, and SMI # will be generated. This bit will be set regardless of whether interrupts or SCI enabled. The SMI# may occur concurrently with an interrupt or SCI.

Figure 90. Power Management PME SMI / SCI / INTR Logic



30.3.3 SERR# Generation

SERR# may be generated via two paths – through PCI mechanisms involving bits in the PCI header, or through PCI Express mechanisms involving bits in the PCI Express capability structure.

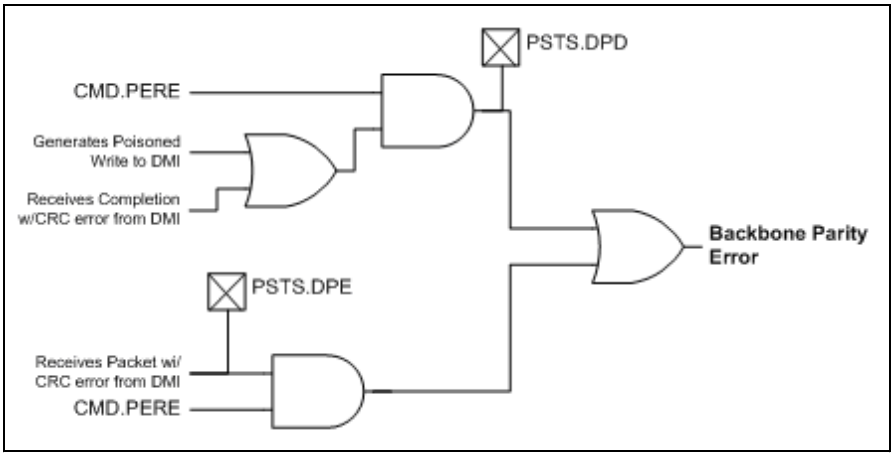
30.3.3.1 PCI Based

30.3.3.1.1 Parity Error (Primary and Secondary)

The root port captures generic data parity errors as well as errors returned on cycles where the root port was the master. If either of these two conditions is met, and the primary side of the bridge is enabled for parity error response, SERR# will be captured.

PSTS.DPD is set when CMD.PERE is set, and receives a completion with parity error or the root port forwards a cycle with bad parity (data poisoning). PSTS.DPE is set when a packet with bad parity is received.

Figure 91. SERR# for Backbone Parity Errors

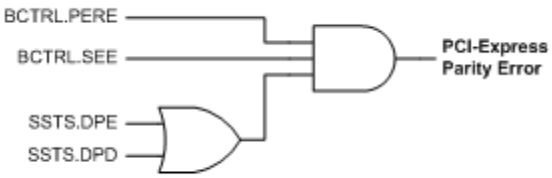


The root port captures generic data parity errors (errors it finds on PCI Express) as well as errors returned on PCI Express cycles where the root port was the master. If either of these two conditions is met, and the secondary side of the bridge is enabled for parity error response, SERR# will be captured as shown in Figure 92.

SSTS.DPD is set when BCTRL.PERE is set, and a poisoned completion is received or the root port is forwarding a bad write (data poisoning). SSTS.DPE is set when a poisoned TLP is received.

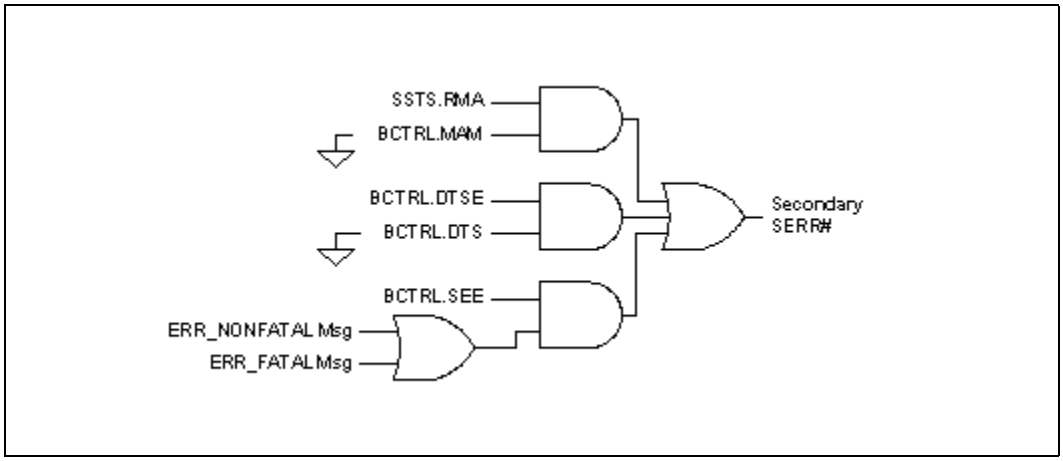
30.3.3.1.2 Secondary Bus Errors

Figure 92. SERR# for Root Port Parity Errors



The PCI bridge specification contains several errors that may occur on the secondary interface. Two of these error types, “master abort mode”, and “delayed transaction discard”, are disallowed per the *PCI Express Base Specification*. The remaining error, SERR# received, is generated when a fatal or non-fatal message is received. These error cases are shown in Figure 93.

Figure 93. SERR# for Secondary Bus Errors



30.3.3.2 PCI Express Based

For the PCI Express based method for generating SERR#, there are three categories: Correctable, Uncorrectable (non-fatal), and Uncorrectable (Fatal). For the root port, these errors are all calculated “looking down” at the PCI Express interface.

Additionally, two sets of enables are used. The DCTL register is used for to enable the root port to generate an error based on looking at the PCI Express interface. What this means is that the root port may determine whether or not there is an error on the packet if the device thought it transmitted a good packet. The RCTL register is used to determine whether a calculated error or received error message should generate SERR#. The following figures graphically show these three error types.

When an error is received or calculated (if enabled), the ESID register is updated with the source. For error messages, this is the requester ID of the sender. For calculated errors, this is the root port’s requester ID.

30.3.3.3 Putting it All Together

Figure 94. SERR# for PCI Express Correctable Errors

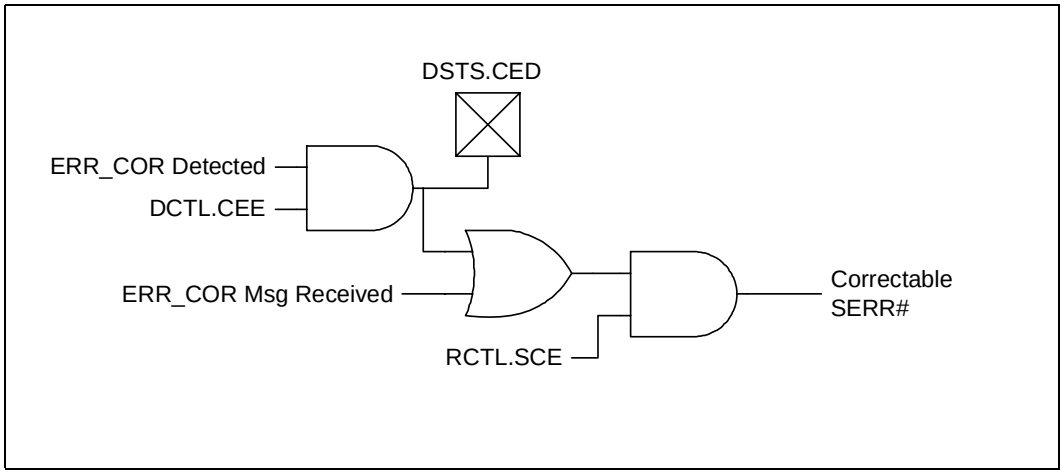


Figure 95. SERR# for PCI Express Uncorrectable (Non-Fatal) Errors

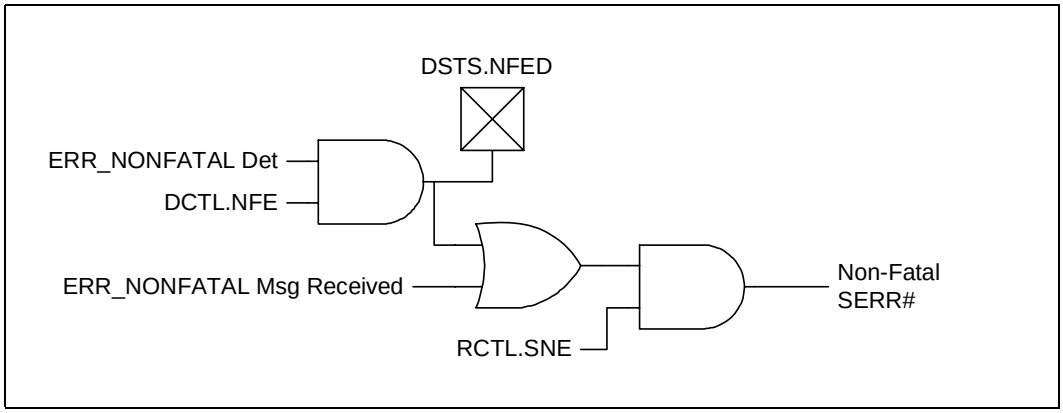
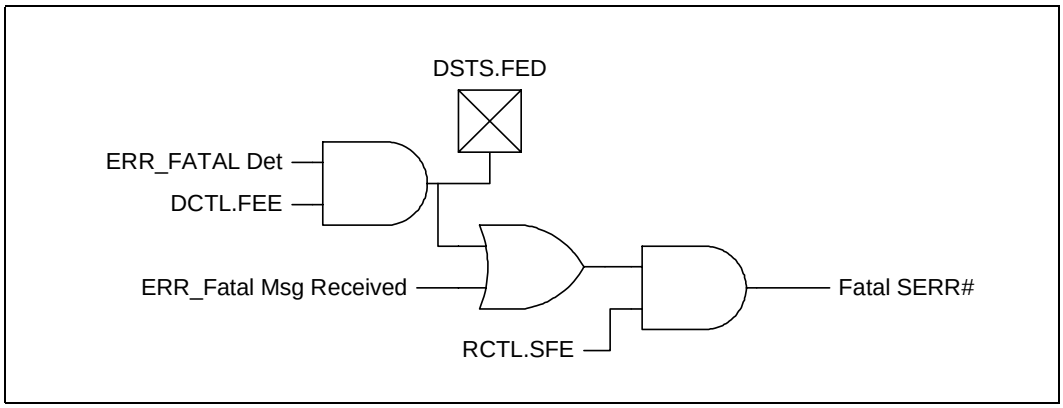
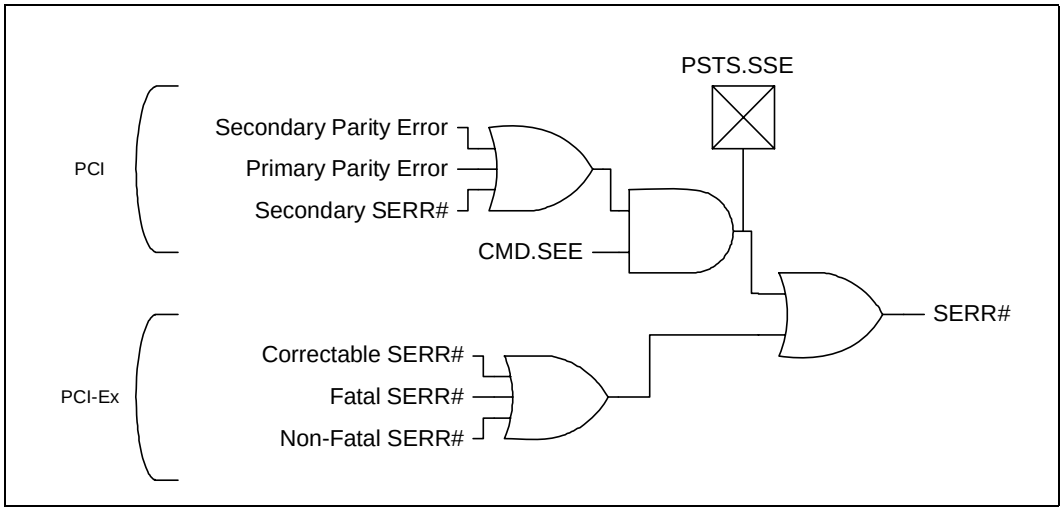


Figure 96. SERR# for PCI Express Uncorrectable (Fatal) Errors



The above two sections described how SERR# may be calculated via register bits in PCI and PCI Express header spaces. These errors are put together to signal SERR# to the platform as shown Figure 97. As can be seen, the PCI Express methods for calculating SERR# do not set PSTS.SSE.

Figure 97. Generation of SERR# to Platform



30.3.4 Hot Plug

Each root port implements a hot plug controller which performs the following:

- Messages to turn on / off / blink LEDs
- Presence and attention button detection
- Interrupt generation

The root port only allows hot plug with modules. Edge-connector based hot plug is not supported.

30.3.4.1 Presence Detection

When a module is plugged in and power is supplied, the physical layer will detect the presence of the device, and the root port sets SLSTS.PDS and SLSTS.PDC. If SLCTL.PDE and SLCTL.HPE are both set, the root port will also generate an interrupt.

When a module is removed (via the physical layer detection), the root port clears SLSTS.PDS and sets SLSTS.PDC. If SLCTL.PDE and SLCTL.HPE are both set, the root port will also generate an interrupt.

The interrupt is generated on an edge-event. For example, if SLSTS.PDC is already set, a change in SLSTS.PDS will not generate a new interrupt. Only SLSTS.PDC going from ‘0’ to ‘1’ will cause an interrupt to be generated.

30.3.4.2 Message Generation

When system software writes to SLCTL.AIC or SLCTL.PIC, the root port will send a message down the link to change the state of LEDs on the module.

Writes to these fields are non-postable cycles, and the resulting message is a postable cycle. When receiving one of these writes, the root port performs the following:

- Changes the state in the register.
- Generates a completion into the upstream queue.
- Formulates a message for the downstream port, if the field is written to regardless of if the field changed.

- Generates the message on the downstream port.
- When the last message of a command is transmitted, sets SLSTS.CC to indicate the command has completed. If SLCTL.CCE and SLCTL.HPE are set, the root port generates an interrupt.

The command completed register applies only to commands issued by software to control the Attention Indicator, Power Indicator or power controller. However, writes to other parts of the Slot Control Register would invariably end up writing to the indicators, power controller fields; Hence, any write to the Slot Control Register is considered a command and if enabled will result in a command complete interrupt. The only exception to this rule is a write to disable the command complete interrupt which must not result in a command complete interrupt.

A single write to the Slot Control register is considered to be a single command, and hence receives a single command complete, even if the write affects more than one field in the Slot Control Register.

30.3.4.3 Attention Button Detection

When an attached device is ejected, an attention button could be pressed by the user. This attention button press will result in the PCI Express message "Attention_Button_Pressed" from the device. Upon receiving this message, the root port will set SLSTS.ABP.

If SLCTL.ABE and SLCTL.HPE are set, the hot plug controller will also generate an interrupt. The interrupt is generated on an edge event. For example, if SLSTS.ABP is already set, a new interrupt will not be generated.

30.3.4.4 SMI/SCI Generation During Hot Plug Event

Interrupts for hot plug events are not supported on legacy operating systems. To support hot plug on non-PCI Express aware operating systems, hot plug events can be routed to generate SCI. To generate SCI, MPC.HPCE must be set. When set, enabled hot plug events will cause SMSCS.HPCS to be set.

Additionally, BIOS workarounds for hot plug can be supported by setting MPC.HPME. When this bit is set, hot plug events can cause SMI status bits in SMSCS to be set. Supported hot plug events and their corresponding SMSCS bit are:

- Command Completed ⇒ SMSCS.HPCCM
- Presence Detect Changed ⇒ SMSCS.HPPDM
- Attention Button Pressed ⇒ SMSCS.HPABM

When any of these bits are set, SMI # will be generated. These bits are set regardless of whether interrupts or SCI is enabled for hot plug events. The SMI# may occur concurrently with an interrupt or SCI.

30.3.4.5 PCI Express Root Port Behavior Clarifications

30.3.4.5.1 Handling of Acks and Naks with certain sequence numbers, received during a replay

Once a replay is initiated by the IICH transmit link layer, it ignores any received Ack or Nak DLLPs that are received during the time that it is replaying previously transmitted TLPs, if the sequence number of the received DLLP references a TLP that was transmitted earlier, but not yet re-transmitted during the current replay. This means:

1. The replay will complete even if a received DLLP Acks a TLP that has yet to be retransmitted.

2. Any TLPs that have been acknowledged by this DLLP will remain in the retry buffer. IICH will need another Ack or Nak to be received before they are purged.
3. The replay timer will not be reset by the reception of these DLLPs

30.3.4.5.2 Hot Reset and Platform Reset

The IICH initiates a Hot Reset (Initiated through a training sequence) only when the Secondary Bus Reset bit is set on the IICH root port's Bridge Control Register (Offset 3Eh), and not based upon PLTRST# assertion. The reset pin on a PCI express card slot, PERST#, needs to be connected to the Platform Reset, PLTRST#, signal (and not the PWROK signal) to ensure that the device is reset under all conditions that the IICH is reset. The PLTRST# signal is the same as the Fundamental Reset signal described in the *PCI Express Base Specification*. In a cabled environment, the PLTRST# signal will need to be delivered to any attached component along with the PCI Express transmit and receive differential signal pairs.

This page intentionally left blank.

31.0 Serial I/O Unit and Watchdog Timer

31.1 Overview

The Serial I/O and Watchdog Timer (SIW) unit is similar to currently available Super I/O controllers. It is specifically designed for integration into the IICH. It is connected via the LPC bus and currently consists of two UARTs, a Serial Interrupt Controller, a Watchdog Timer and the LPC interface.

31.2 Features

LPC Interface

- Multiplexed command, address and data bus
- 8-bit I/O transfers
- 16-bit address qualification for I/O transactions
- Serial IRQ interface compatible with serialized IRQ support for PCI systems

Serial Port

- Two Full Function 16550 Compatible Serial Ports
- Configurable I/O addresses and interrupts
- 16-Byte FIFOs
- Supports up to 115 Kbps
- Programmable Baud Rate Generator
- Modem Control Circuitry
- 14.7456 MHz, 33 MHz, and 48 MHz supported for UART baud clock input

Watchdog Timer (WDT)

Selectable Prescaler – approximately 1 MHz (1 µs to 1 s) and approximately 1 KHz (1 ms to 1050 s)

- 33 MHz Clock (30 ns Clock Ticks)
- Multiple Modes (WDT and Free-Running)
- Free-Running Mode:
  - 1 Stage Timer - Toggles WDT_TOUT# after programmable time.
- WDT Mode:
  - 2 Stage Timer (First stage generates interrupt, second stage drives WDT_TOUT# low)
- First stage generates an SERIRQ, SMI or NMI interrupt (depending on which is enabled) after the programmable time is reached.
- Second stage drives WDT_OUT# low or inverts the previous value.

— Used only after first timeout occurs.

— Status bit preserved in RTC well for possible error detection and correction.

— Drives WDT_TOUT# if OUTPUT is enabled.

- Timer can be disabled (default state) or Locked (Hard Reset required to disable WDT)
- When WDT Reload Sequence is performed the WDT Automatic Reload of Preload value occurs

31.3 Signal Description

Signal inputs and outputs for the SIW are shown in Table 1041.

Table 1041. SIW Pin List (Sheet 1 of 2)

Signal Name	Type	Description
UART_CLK	I	**Input clock to the SIU:** This clock is passed to the baud clock generation logic of each UART in the SIU.
SIU1_RXD, SIU2_RXD	I	**SERIAL INPUT for UART1 and UART2:** Serial data input from device pin to the receive port.
SIU1_TXD, SIU2_TXD,	O	**SERIAL OUTPUT for UART1 and UART2:** Serial data output to the communication peripheral/modem or data set. Upon reset, the TXD pins are set to MARKING condition (logic 1).
SIU1_CTS#, SIU2_CTS#,	I	**CLEAR TO SEND for UART1 and UART2:** Active low, this pin indicates that data can be exchanged between the Intel® 3100 Chipset and external interface. These pins have no effect on the transmitter. These pins can used as Modem Status Input whose condition can be tested by the processor by reading bit 4 (CTS) of the Modem Status register (MSR). Bit 4 is the complement of the CTS# signal. Bit 0 (DCTS) of the MSR indicates whether the CTS# input has changed state since the previous reading of the MSR. When the CTS bit of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.
SIU1_DSR#, SIU2_DSR#	I	**DATA SET READY for UART1 and UART2:** Active low, this pin indicates that the external agent is ready to communicate with the Intel® 3100 ChipsetUART. These pins have no effect on the transmitter. These pins could be used as Modem Status Input whose condition can be tested by the processor by reading bit 5 (DSR) of the Modem Status register. Bit 5 is the complement of the DSR# signal. Bit 1 (DDSR) of the Modem status register (MSR) indicates whether the DSR# input has changed state since the previous reading of the MSR. When the DSR bit of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.
SIU1_DCD#, SIU2_DCD#,	I	**DATA CARRIER DETECT for UART1 and UART2:** Active low, this pin indicates that data carrier has been detected by the external agent. These pins are Modem Status Input whose condition can be tested by the processor by reading bit 7 (DCD) of the Modem Status register (MSR). Bit 7 is the complement of the DCD# signal. Bit 3 (DDCD) of the MSR indicates whether the DCD# input has changed state since the previous reading of the MSR. When the DCD bit of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.
SIU1_RI#, SIU2_RI#,	I	**RING INDICATOR for UART1 and UART2:** Active low, this pin indicates that a telephone ringing signal has been received by the external agent. These pins are Modem Status Input whose condition can be tested by the processor by reading bit 6 (RI) of the Modem Status register (MSR). Bit 6 is the complement of the RI# signal. Bit 2 (TERI) of the MSR indicates whether the DCD# input has changed state since the previous reading of the MSR. When the RI bit of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.

Table 1041. SIW Pin List (Sheet 2 of 2)

Signal Name	Type	Description
SIU1_DTR#, SIU2_DTR#	O	**DATA TERMINAL READY for UART1 and UART2:** When low these pins inform the modem or data set that the Intel® 3100 Chipset UART 1, 2 are ready to establish a communication link. The DTR#X(X=0,1) output signals can be set to an active low by programming the DTRX (X-0,1) (bit0) of the Modem control register to a logic 1. A Reset operation sets this signal to its inactive state (logic 1). LOOP mode operation holds this signal in its inactive state.
SIU1_RTS#, SIU2_RTS#,	O	**REQUEST TO SEND for UART1 and UART2:** When low these pins informs the modem or data set that the Intel® 3100 Chipset UART 1, 2 are ready to establish a communication link. The RTS#X(X=0,1) output signals can be set to an active low by programming the RTS X (X-0,1) (bit1) of the Modem control register to a logic 1. A Reset operation sets this signal to its inactive state (logic 1). LOOP mode operation holds this signal in its inactive state.
WDT_TOUT#	O	**Watchdog Timer Timeout:** The WDT_TOUT# signal is driven low from the IICH to an external pin. The signal is driven low when the main 35-bit down counter reaches zero during the seconds stage. The WDT_TOUT_CNF bit in the WDT Lock register determines if the output is to change from the previous state if another time out occurs, or WDT_TOUT# is driven low until the system is reset or power is cycled.

31.4 Functional Description

31.4.1 Host Processor Interface (LPC)

The host processor communicates with the SIW via the LPC bus. Access is through a series of read/ write registers and accomplished through I/O cycles. All registers are eight bits wide. The SIW registers include global configuration space and device specific regions accessed by setting the Logical Device Number in the SIW Configuration Register 07H (SCR7).

Table 1042. Address Map

Address	Block Name	Logical Device
04Eh or 20Eh (SIU1_DTR# dependent)	Configuration Index	
04Fh or 20Fh (SIU1_DTR# dependent)	Configuration Data	
Base+(0-7)	Serial Port 1	04H
Base+(0-7)	Serial Port 2	05H
Base+(0-18)	Watchdog Timer	06H

See Section 31.9 for configuration register descriptions and information on setting the base address.

31.5 LPC Interface

The LPC interface is used to control all the logical blocks on the SIW. LPC bus signals use
PCI 33 MHz electrical signal characteristics. Refer to the *Low Pin Count (LPC) Interface Specification Rev 1.0.*

31.5.1 LPC Cycles

The following cycle types are supported by the LPC protocol.

Table 1043. Supported LPC Cycle Types

Cycle Type	Transfer Size
I/O Write	1 Byte
I/O Read	1 Byte

The SIW ignores cycles that it does not support.

31.5.1.1 I/O Read and Write Cycles

The SIW is the target for I/O cycles. I/O cycles are initiated by the host for register or FIFO accesses and generally have minimal synchronization times.

Data transfers are assumed to be exactly 1-byte. If the CPU requested a 16-bit or 32-bit transfer, the host must break it up into 8-bit transfers.

See the *LPC Interface Specification* for the sequence of cycles for the I/O Read and Write cycles.

31.5.2 Policy

The following rules govern the reset policy:

SIW_RESET# is tied to the internal PCI bus reset.

When SIW_RESET# goes active (low):

- The host drives the LFRAME# signal high, tri-states the LAD[3:0] signals.
- The SIU ignores LFRAME#, tri-states the LAD[3:0] pins.

Note: LPC bus signals from SIW are internally tied to the primary LPC interface of the IICH device. Host LPC and SIW LPC names are used interchangeably throughout.

31.5.3 LPC Transfers

31.5.3.1 I/O Transfers

These are generally used for register or FIFO accesses, and generally have minimal synchronization times. The minimum number of wait-states between bytes is one. Data transfers are assumed to be exactly one byte. The host is responsible for breaking up larger data transfers into 8-bit cycles.

Table 1044. I/O Sync Bits Description

Bits	Indication
0000	Synchronization achieved with no error.
0101	Indicates that synchronization not achieved yet, but the part is driving the bus.
0110	Indicates that synchronization not achieved yet, but the part is driving the bus and expects long synchronization
1010	Special Case: peripheral indicating errors.

31.6 Logical Devices 4 and 5: Serial Ports (UART1 and UART2)

This section describes the Universal Asynchronous Receiver/Transmitter (UART) serial port used for the two UART integrated into the SIW. The UART can be controlled via programmed I/O. The basic programming model is the same for both UARTs with the only difference being the Logical Device Number assigned to each.

The serial port consists of a UART which supports all the functions of a standard 16550 UART including hardware flow control interface.

The UART performs serial-to-parallel conversion on data characters received from a peripheral device or a modem and parallel-to-serial conversion on data characters received from the processor. The processor can read the complete status of the UART at any time during the functional operation. Available status information includes the type and condition of the transfer operations being performed by the UART and any error conditions (parity, overrun, framing, or break interrupt).

The serial port can operate in either FIFO or non-FIFO mode. In FIFO mode, a 16-byte transmit FIFO holds data from the processor to be transmitted on the serial link and a 16-byte Receive FIFO buffers data from the serial link until read by the processor.

Each UART includes a programmable baud rate generator which is capable of dividing the baud clock input by divisors of one to (2^{16} -1) and producing a 16X clock to drive the internal transmitter and receiver logic. Each UART has complete modem control capability and a processor interrupt system. Interrupts can be programmed to the user's requirements, minimizing the computing required to handle the communications link. Each UART can operate in a polled or an interrupt driven environment as configured by software.

The baud rate generator input is a function of the UART_CLK and a configurable predivide of 1, 8, or 26. See also SIW Configuration (address 29h) in Section 31.9.3.1, “Global Control/Configuration Registers [00h - 2Fh]” on page 1047. The output of the baud rate generator is 16 times the baud rate.

Table 1045. UART Clock Divider Support

Clock Frequency	14.7456 MHz	48.0 MHz
Predivide Value	8	26
Generator Frequency	1.8432 MHz	1.8462 MHz

Table 1046. Baud Rate Example

Desired Baud Rate	Divisor	% error @ 1.8432	% error @ 1.8462*
300	384		0.16
1200	96		0.16
2400	48		0.16
4800	24		0.16
9600	12		0.16
19200	6		0.16
38400	3		0.16
56000	2	2.8	3
115200	1		0.16

31.6.1 UART Feature List

- Functionally compatible with National Semiconductor's PC16550D
- Adds or deletes standard asynchronous communications bits (start, stop, and parity) to or from the serial data
- Independently controlled transmit, receive, line status and data set interrupts
- Programmable baud rate generator allows division of clock by 1 to (2^{16} -1) and generates an internal 16X clock
- Modem control functions (CTS#, RTS#, DSR#, DTR#, RI#, and DCD#)
- Fully programmable serial-interface characteristics:
- 5, 6, 7 or 8-bit characters
- Even, odd, or no parity detection
- 1, 1-1/2, or 2 stop bit generation
- Baud rate generation (up to 115 kbps)
- False start bit detection
- 16-byte Receive FIFO
- Complete status reporting capability
- Line break generation and detection
- Internal diagnostic capabilities include:
  - — Loopback controls for communications link fault isolation
  - — Break, parity, overrun, and framing error simulation
  - — Fully prioritized interrupt system controls

31.6.2 Signal Descriptions

The name and description of the UART sub-block signals are shown in Table 1047.

Table 1047. UART Signal Descriptions (Sheet 1 of 2)

Name	Type	Description
rxd	Input	**SERIAL INPUT:** Serial data input from device pin to the receive shift register.
txd	Output	**SERIAL OUTPUT:** Composite serial data output to the communications link-peripheral, modem, or data set. The TXD signal is set to the MARKING (logic 1) state upon a Reset operation.
cts_n	Input	**CLEAR TO SEND:** When low, this pin indicates that the modem or data set is ready to exchange data. The CTS# signal is a modem status input whose condition can be tested by the processor when reading bit 4 (CTS) of the Modem Status register. Bit 4 is the complement of the CTS# signal. Bit 0 (DCTS) of the Modem Status register indicates whether the CTS# input has changed state since the previous reading of the Modem Status register. CTS# has no effect on the transmitter. When the CTS bit of the Modem Status register changes state, an interrupt is generated if the Modem Status interrupt is enabled.
dsr_n	Input	**DATA SET READY:** When low, this pin indicates that the modem or data set is ready to establish the communications link with the UART. The DSR# signal is a Modem Status input whose condition can be tested by the processor by reading bit 5 (DSR) of the Modem Status register. Bit 5 is the complement of the DSR# signal. Bit 1 (DDSR) of the Modem Status register indicates whether the DSR# input has changed state since the previous reading of the Modem Status register. DSR# has no effect on the transmitter. When the DSR bit of the Modem Status register changes state, an interrupt is generated if the Modem Status interrupt is enabled.

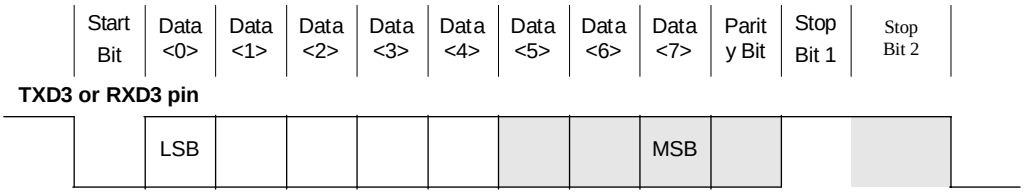
Table 1047. UART Signal Descriptions (Sheet 2 of 2)

Name	Type	Description
dcd_n	Input	**DATA CARRIER DETECT:** When low, this pin indicates that the data carrier has been detected by the modem or data set. The DCD# signal is a modem status input whose condition can be tested by the processor by reading bit 7 (DCD) of the Modem Status register. Bit 7 is the complement of the DCD# signal. Bit 3 (DDCD) of the Modem Status register indicates whether the DCD# input has changed state since the previous reading of the Modem Status register. DCD# has no effect on the receiver. When the DCD bit changes state, an interrupt is generated if the Modem Status interrupt is enabled.
ri_n	Input	**RING INDICATOR:** When low, this pin indicates that a telephone ringing signal has been received by the modem or data set. The RI# signal is a Modem Status input whose condition can be tested by the processor when reading bit 6 (RI) of the Modem Status register. Bit 6 is the complement of the RI# signal. Bit 2 (TERI), the Trailing Edge of Ring Indicator, of the Modem Status register indicates whether the RI# input signal has changed from low to high since the previous reading of the Modem Status register. When the RI bit of the Modem status register changes from a high to low state, an interrupt is generated if the Modem status interrupt is enabled.
dtr_n	Output	**DATA TERMINAL READY:** When low, this pin informs the modem or data set that the UART is ready to establish a communications link. The DTR# output signal can be set to an active low by programming bit 0 (DTR) of the Modem Control register to a 1. A Reset operation sets this signal to its inactive state. LOOP mode operation holds this signal in its inactive state.
rts_n	Output	**REQUEST TO SEND:** When low, this informs the modem or the data set that the UART is ready to exchange data. The RTS# output signal can be set to an active low by programming bit 1 (RTS) of the Modem Control register to a 1. A Reset operation sets this signal to its inactive (high) state. LOOP mode operation holds this signal in its inactive state.
clk_uart	Input	**BAUD CLOCK INPUT:** Clock input to the Baud Rate Generator.
int	Output	**INTERRUPT:** This pin goes high when any one of the following interrupt types has an active high condition and is enabled via the IER register: Receiver Error Flag, Received Data Available, Transmit Holding Register Empty, Timeout in FIFO mode and Modem Status. The int signal is reset low upon the appropriate interrupt service or Reset operation

31.6.3 UART Operational Description

The format of a UART data frame is shown in Figure 98.

Figure 98. Example UART Data Frame



Receive data sample counter frequency = 16x bit frequency, each bit is sampled three times in the middle. Shaded bits are optional and can be programmed by users.

Each data frame is between seven bits and 12 bits long depending on the size of data programmed, if parity is enabled and if two stop bits is selected. The frame begins with a start bit that is represented by a high to low transition. Next, 5 to 8 bits of data are

transmitted, beginning with the least significant bit. An optional parity bit follows, which is set if even parity is enabled and an odd number of ones exist within the data byte, or if odd parity is enabled and the data byte contains an even number of ones. The data frame ends with one, one and a half or two stop bits as programmed by the user, which is represented by one or two successive bit periods of a logic one.

The unit is disabled upon reset, the user needs to enable the unit by setting bit six of Interrupt Enable Register. When the unit is enabled, the receiver starts looking for the start bit of a frame; the transmitter starts transmitting data to the transmit data pin if there is data available in the transmit FIFO. Transmit data can be written to the FIFO before the unit is enabled. When the unit is disabled, the transmitter/receiver finishes the current byte being transmitted/received if it is in the middle of transmitting/ receiving a byte and stops transmitting/receiving more data.

An SIU_RESET# to the SIU forces the internal register and output signals on the serial port to the values listed below.

Table 1048. UART Register/Signal Reset States

Register/Signal	Reset Control	Reset State
Interrupt Enable Register	RESET	All bits are low.
Interrupt ID Register	RESET	Bit 0 is forced high. Bits 1-3 and 6-7 are forced low. Bits 4-5 are permanently low.
Line Control Register	RESET	All bits are forced low.
Line Status Register	RESET	Bits 0-4,7 are forced low. Bits 5 and 6 are forced high.
Modem Control Register	RESET	Bits 0,1,2,3,4 are forced low. Bits 5,6,7 are permanently low.
Modem Status Register	RESET/Modem signal, read MSR for bits 3-0.	Low
Infrared Selection Register	RESET	All bits are permanently low.
Txd	RESET	High
Int	RESET/ clear LINE STATUS REG	Low
rts_n	RESET	High
dtr_n	RESET	High

31.6.4 Internal Register Descriptions

There are 12 registers in the UART. These registers share eight address locations in the I/O address space. Table 1049 shows the registers and their addresses as offsets of a base address. The state of the Divisor Latch Bit (DLAB), which is the MOST significant bit of the Serial Line Control Register, affects the selection of certain of the UART registers. The DLAB bit must be set high by the system software to access the Baud Rate Generator Divisor Latches.

Table 1049. Internal Register Descriptions (Sheet 1 of 2)

UART Register Addresses (Base + offset)	DLAB Bit Value	Register Accessed
Base	0	Receive BUFFER (Read-Only)
Base	0	Transmit BUFFER (Write-Only)
Base + 01H	0	Interrupt Enable (Read/Write)
Base + 02H	X	Interrupt I.D. (Read-Only)

Table 1049. Internal Register Descriptions (Sheet 2 of 2)

Base + 02H	X	FIFO Control (Write-Only)
Base + 03H	X	Line Control (Read/Write)
Base + 04H	X	Modem Control (Read/Write)
Base + 05H	X	Line Status (Read-Only)
Base + 06H	X	Modem Status (Read-Only)
Base + 07H	X	Scratch Pad (Read/Write)
Base	1	Divisor Latch (Lower Byte, Read/Write)
Base + 01H	1	Divisor Latch (Upper Byte, Read/Write)

31.6.4.1 RBR – Receive Buffer Register

In non-FIFO mode, this register holds the character received by the UART's Receive Shift Register. If fewer than eight bits are received, the bits are right-justified and the leading bits are zeroed. Reading the register empties the register and resets the Data Ready (DR) bit in the Line Status Register to zero. Other (error) bits in the Line Status Register are not cleared. In FIFO mode, this register latches the value of the data byte at the top of the FIFO.

Table 1050. RBR – Receive Buffer Register

Offset: Base (DLAB=0) ***Size:*** 8 bit
Default Value: 00h ***Power Well:*** *Core*
Lockable: *No*

Bits	Name	Description	Reset Value	Access
07:00	RB[7:0]	Data byte received, least significant bit first	00h	RO

31.6.4.2 THR – Transmit Holding Register

This register holds the next data byte to be transmitted. When the Transmit Shift Register becomes empty, the contents of the Transmit Holding Register are loaded into the shift register and the transmit data request (TDRQ) bit in the Line Status Register is set to one.

Table 1051. THR – Transmit Holding Register

Offset: Base (DLAB=0) ***Size:*** *8 bit*
Default Value: *00h* ***Power Well:*** *Core*
Lockable: *No*

Bits	Name	Description	Reset Value	Access
07:00	TB[7:0]	Data byte transmitted, least significant bit first.	00h	W

In FIFO mode, writing to THR puts data to the top of the FIFO. The data at the bottom of the FIFO is loaded to the shift register when it is empty.

31.6.4.3 IER – Interrupt Enable Register

This register enables five types of interrupts which independently activate the int signal and set a value in the Interrupt Identification Register. Each of the five interrupt types can be disabled by resetting the appropriate bit of the IER register. Similarly, by setting

the appropriate bits, selected interrupts can be enabled. Receiver time out interrupt can be configured to be separated from the receive data available interrupt (using the bit 5: COMP)

The use of bit 4 and 5 is different from the register definition of standard 16550.

Table 1052. IER – Interrupt Enable Register

Offset: Base + 01H (DLAB=0)	**Size:** 8 bit
Default Value: 00h	**Power Well:** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07:06	RSVD	RSVD = 0	00	RO
05	COMP	Compatibility Enable: 0 = Bit 0 of this register also controls RTOIE and bit 4 is RSVD. 1 = Bit 4 of this register controls RTOIE.	0	RW
04	RTOIE	Receiver Time Out Interrupt Enable: 0 = Receiver data Time out interrupt disabled 1 = Receiver data Time out interrupt enabled	0	RW
03	MIE	Modem Interrupt Enable: 0 = Modem Status interrupt disabled 1 = Modem Status interrupt enabled	0	RW
02	RLSE	Receiver Line Status Interrupt Enable: 0 = Receiver Line Status interrupt disabled 1 = Receiver Line Status interrupt enabled	0	RW
01	TIE	Transmit Data request Interrupt Enable: 0 = Transmit FIFO Data Request interrupt disabled 1 = Transmit FIFO Data Request interrupt enabled	0	RW
00	RAVIE	Receiver Data Available Interrupt Enable: When BIT 5 = 1: 0 = Receiver Data Available (Trigger level reached) interrupt disabled 1 = Receiver Data Available (Trigger level reached) interrupt enabled When BIT 5 = 0: 0 = Receiver data Time Out Interrupt also disabled 1 = Receiver data Time Out Interrupt enabled	0	RW

31.6.4.4 IIR – Interrupt Identification Register

In order to minimize software overhead during data character transfers, the UART prioritizes interrupts into four levels (listed in Table 1053) and records these in the Interrupt Identification Register. The Interrupt Identification Register (IIR) stores information indicating that a prioritized interrupt is pending and the source of that interrupt.

Table 1053. Interrupt Conditions (Sheet 1 of 2)

Priority Level	Interrupt Origin
1 (highest)	Receiver Line Status. One or more error bits were set.
2	Received Data is available. In FIFO mode, trigger level was reached; in non-FIFO mode, RBR has data.

Table 1053. Interrupt Conditions (Sheet 2 of 2)

2	Receiver Time out occurred. It happens in FIFO mode only, when there is data in the receive FIFO but no activity for a time period.
3	Transmitter requests data. In FIFO mode, the transmit FIFO is half or more than half empty; in non-FIFO mode, THR is read already.
4	Modem Status: one or more of the modem input signals has changed state

Table 1054. IIR – Interrupt Identification Register

Offset: *Base + 02H* **Size:** *8 bit*
Default Value: *01h* **Power Well:** *Core*
Lockable: *No*

Bits	Name	Description	Reset Value	Access
07:06	FIFOES[1:0]	FIFO Mode Enable Status: 00 Non-FIFO mode is selected. 01 Reserved 10 Reserved 11 FIFO mode is selected (TRFIFOE = 1).	00b	RO
05:04	Reserved	Reserved	00b	
03	TOD (IID3)	Time Out Detected: 0 = No time out interrupt is pending. 1 = Time out interrupt is pending. (FIFO mode only)	0b	RO
02:01	IID[2:1]	Interrupt Source Encoded: 00 Modem Status (CTS, DSR, RI, DCD modem signals changed state) 01 Transmit FIFO requests data 10 Received Data Available 11 Receive error (Overrun, parity, framing, break, FIFO error)	00b	RO
00	P#	Interrupt Pending: 0 = Interrupt is pending. (Active low) 1 = No interrupt is pending.	1b	RO

Table 1055. Interrupt Identification Register Decode (Sheet 1 of 2)

Interrupt ID bits				Interrupt SET/RESET Function			
3	2	1	0	Priority	Type	Source	RESET Control
0	0	0	1	-	None	No Interrupt is pending.	-
0	1	1	0	Highest	Receiver Line Status	Overrun Error, Parity Error, Framing Error, Break Interrupt.	Reading the Line Status Register.
0	1	0	0	Second Highest	Received Data Available.	Non-FIFO mode: Receive Buffer is full.	Non-FIFO mode: Reading the Receiver Buffer Register.
						FIFO mode: Trigger level was reached.	FIFO mode: Reading bytes until Receiver FIFO drops below trigger level or setting RESETRF bit in FCR register.
1	1	0	0	Second Highest	Character Timeout indication.	FIFO Mode only: At least 1 character is in receiver FIFO and there was no activity for a time period.	Reading the Receiver FIFO or setting RESETRF bit in FCR register.

Table 1055. Interrupt Identification Register Decode (Sheet 2 of 2)

0	0	1	0	Third Highest	Transmit FIFO Data Request	Non-FIFO mode: Transmit Holding Register Empty	Reading the IIR Register (if the source of the interrupt) or writing into the Transmit Holding Register.
						FIFO mode: Transmit FIFO has half or less than half data.	Reading the IIR Register (if the source of the interrupt) or writing to the Transmitter FIFO.
0	0	0	0	Fourth Highest	Modem Status	Clear to Send, Data Set Ready, Ring Indicator, Received Line Signal Detect	Reading the modem status register

31.6.4.5 FCR – FIFO Control Register

FCR is a write-only register that is located at the same address as the IIR (IIR is a read-only register). FCR enables/disables the transmitter/receiver FIFOs, clears the transmitter/receiver FIFOs, and sets the receiver FIFO trigger level.

The use of bit 6 and 7 is different from the register definition of standard 16550

Table 1056. FCR – FIFO Control Register (Sheet 1 of 2)

Offset: *Base + 02H* ***Size:*** *8 bit*
Default Value: *00h* ***Power Well:*** *Core*
Lockable: *No*

Bits	Name	Description	Reset Value	Access
07:06	ITL[1:0]	**Interrupt Trigger Level:** When the number of bytes in the receiver FIFO equals the interrupt trigger level programmed into this field and the Received Data Available Interrupt is enabled (via IER), an interrupt is generated and appropriate bits are set in the IIR. 00 1 byte or more in FIFO causes interrupt 01 RSVD 10 8 bytes or more in FIFO causes interrupt 11 RSVD	00b	WO
05:03	Reserved	Reserved. Must be programmed to 0.	000b	

Table 1056. FCR – FIFO Control Register (Sheet 2 of 2)

Offset: Base + 02H	**Size:** 8 bit
Default Value: 00h	**Power Well:** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
02	RESETTF	**Reset transmitter FIFO:** When RESETTF is set to 1, the transmitter FIFO counter logic is set to 0, effectively clearing all the bytes in the FIFO. The TDRQ bit in LSR are set and IIR shows a transmitter requests data interrupt if the TIE bit in the IER register is set. The transmitter shift register is not cleared; it completes the current transmission. After the FIFO is cleared, RESETTF is automatically reset to 0. 0 = Writing 0 has no effect 1 = The transmitter FIFO is cleared (FIFO counter set to 0). After clearing, bit is automatically reset to 0	0b	WO
01	RESETRF	**Reset Receiver FIFO:** When RESETRF is set to 1, the receiver FIFO counter is reset to 0, effectively clearing all the bytes in the FIFO. The DR bit in LSR is reset to 0. All the error bits in the FIFO and the FIFOE bit in LSR are cleared. Any error bits, OE, PE, FE or BI, that had been set in LSR are still set. The receiver shift register is not cleared. If IIR had been set to Received Data Available, it is cleared. After the FIFO is cleared, RESETRF is automatically reset to 0. 0 = Writing 0 has no effect 1 = The receiver FIFO is cleared (FIFO counter set to 0). After clearing, bit is automatically reset to 0	0b	WO
00	TRFIFOE	**Transmit and Receive FIFO Enable:** TRFIFOE enables/ disables the transmitter and receiver FIFOs. When TRFIFOE = 1, both FIFOs are enabled (FIFO Mode). When TRFIFOE = 0, the FIFOs are both disabled (non-FIFO Mode). Writing a 0 to this bit clears all bytes in both FIFOs. When changing from FIFO mode to non-FIFO mode and vice versa, data is automatically cleared from the FIFOs. This bit must be 1 when other bits in this register are written or the other bits are not programmed. 0 = FIFOs are disabled 1 = FIFOs are enabled	0b	WO

31.6.4.6 LCR – Line Control Register

In the Line Control Register (LCR), the system programmer specifies the format of the asynchronous data communications exchange. The serial data format consists of a start bit (logic 0), five to eight data bits, an optional parity bit, and one or two stop bits (logic 1). The LCR has bits for accessing the Divisor Latch and causing a break condition. The programmer can also read the contents of the Line Control Register. The read capability simplifies system programming and eliminates the need for separate storage in system memory.

Table 1057. LCR – Line Control Register (Sheet 1 of 2)

Offset: Base + 03H	**Size:** 8 bit
Default Value: 00h	**Power Well:** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07	DLAB	**Divisor register access bit:** This bit is the Divisor Latch Access Bit. It must be set high (logic 1) to access the Divisor Latches of the Baud Rate Generator during a READ or WRITE operation. It must be set low (logic 0) to access the Receiver Buffer, the Transmit Holding Register, or the Interrupt Enable Register. 0 = Access Transmit Holding register (THR), Receive Buffer Register (RBR) and Interrupt Enable Register. 1 = Access Divisor Latch Registers (DLL and DLM).	0b	RW
06	SB	**Set break:** This bit is the set break control bit. It causes a break condition to be transmitted to the receiving UART. When SB is set to a logic 1, the serial output (TXD) is forced to the spacing (logic 0) state and remains there until SB is set to a logic 0. This bit acts only on the TXD pin and has no effect on the transmitter logic. This feature enables the processor to alert a terminal in a computer communications system. If the following sequence is executed, no erroneous characters are transmitted because of the break: Load 00H in the Transmit Holding register in response to a TDRQ interrupt After TDRQ goes high (indicating that 00H is being shifted out), set the break bit before the parity or stop bits reach the TXD pin Wait for the transmitter to be idle (TEMT = 1) and clear the break bit when normal transmission has to be restored During the break, the transmitter can be used as a character timer to accurately establish the break duration. In FIFO mode, wait for the transmitter to be idle (TEMT=1) to set and clear the break bit. 0 = No effect on TXD output 1 = Forces TXD output to 0 (space)	0b	RW
05	STKYP	**Sticky Parity:** This bit is the “sticky parity” bit, which can be used in multiprocessor communications. When PEN and STKYP are logic 1, the bit that is transmitted in the parity bit location (the bit just before the stop bit) is the complement of the EPS bit. If EPS is 0, then the bit at the parity bit location are transmitted as a 1. In the receiver, if STKYP and PEN are 1, then the receiver compares the bit that is received in the parity bit location with the complement of the EPS bit. If the values being compared are not equal, the receiver sets the Parity Error bit in LSR and causes an error interrupt if line status interrupts were enabled. For example, if EPS is 0, the receiver expects the bit received at the parity bit location to be 1. If it is not, then the parity error bit is set. By forcing the bit value at the parity bit location, rather than calculating a parity value, a system with a master transmitter and multiple receivers can identify some transmitted characters as receiver addresses and the rest of the characters as data. If PEN = 0, STKYP is ignored. 0 = No effect on parity bit 1 = Forces parity bit to be opposite of EPS bit value	0b	RW

Table 1057. LCR – Line Control Register (Sheet 2 of 2)

Offset: Base + 03H	**Size:** 8 bit
Default Value: 00h	**Power Well:** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
04	EPS	**Even parity Select:** This bit is the even parity select bit. When PEN is a logic 1 and EPS is a logic 0, an odd number of logic ones is transmitted or checked in the data word bits and the parity bit. When PEN is a logic 1 and EPS is a logic 1, an even number of logic ones is transmitted or checked in the data word bits and parity bit. If PEN = 0, EPS is ignored. 0 = Sends or checks for odd parity 1 = Sends or checks for even parity	0b	RW
03	PEN	**Parity enable:** This is the parity enable bit. When PEN is a logic 1, a parity bit is generated (transmit data) or checked (receive data) between the last data word bit and Stop bit of the serial data. (The parity bit is used to produce an even or odd number of ones when the data word bits and the parity bit are summed.) 0 = No parity function 1 = Allows parity generation and checking	0b	RW
02	STB	**Stop bits:** This bit specifies the number of stop bits transmitted and received in each serial character. If STB is a logic 0, one stop bit is generated in the transmitted data. If STB is a logic 1 when a 5-bit word length is selected via bits 0 and 1, then 1 and one half stop bits are generated. If STB is a logic 1 when either a 6, 7, or 8-bit word is selected, then two stop bits are generated. The receiver checks the first stop bit only, regardless of the number of stop bits selected. 0 = 1 stop bit 1 = 2 stop bits, except for 5-bit character then 1-1/2 bits	0b	RW
01:00	WLS[1:0]	**Word Length select:** The Word Length Select bits specify the number of data bits in each transmitted or received serial character. 00 5-bit character (default) 01 6-bit character 10 7-bit character 11 8-bit character	00b	RW

31.6.4.7 LSR – Line Status Register

This register provides status information to the processor concerning the data transfers. Bits 5 and 6 show information about the transmitter section. The rest of the bits contain information about the receiver.

In non-FIFO mode, three of the LSR register bits, parity error, framing error, and break interrupt, show the error status of the character that has just been received. In FIFO mode, these three bits of status are stored with each received character in the FIFO. LSR shows the status bits of the character at the top of the FIFO. When the character at the top of the FIFO has errors, the LSR error bits are set and are not cleared until software reads LSR, even if the character in the FIFO is read and a new character is now at the top of the FIFO.

Bits one through four are the error conditions that produce a receiver line status interrupt when any of the corresponding conditions are detected and the interrupt is enabled. These bits are not cleared by reading the erroneous byte from the FIFO or receive buffer. They are cleared only by reading LSR. In FIFO mode, the line status interrupt occurs only when the erroneous byte is at the top of the FIFO. If the

erroneous byte being received is not at the top of the FIFO, an interrupt is generated only after the previous bytes are read and the erroneous byte is moved to the top of the FIFO.

Table 1058. Line Status Register (LSR) (Sheet 1 of 2)

Offset: Base + 05H	**Size:** 8 bit
Default Value: 60h	**Power Well:** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07	FIFOE	**FIFO Error Status:** This bit is reset only when all the error bytes have been read from the FIFO. A processor read to the Line Status register does not reset this bit. Non-FIFO mode: 0 = Bit is always "0" indicating no FIFO. FIFO mode: 0 = All error bytes have been read from the FIFO 1 = At least one character in the receiver FIFO contains a parity error, framing error, or break indication. When DMA requests are enabled (IER bit7 is set to 1) and FIFOE is set to 1, no receive DMA request is generated even though the receive FIFO reaches the trigger level and the error interrupt is generated. When DMA requests are not enabled (IER bit7 is set to 0), FIFOE set to 1 does not generate interrupt.	0b	RO
06	TEMT	Transmitter Empty: Non-FIFO mode: 0 = Either the Transmit Holding register or the Tansmitter Shift register contain a data character. 1 = The Transmit Holding register and the Transmitter Shift register are both empty. FIFO mode: 0 = Either the transmitter FIFO or the Transmit Shift register contain a data character. 1 = The transmitter FIFO and the Transmit Shift register are both empty.	1b	RO
05	TRDQ	**Transmit Data Request:** TDRQ indicates that the UART is ready to accept a new character for transmission. In addition, this bit causes the UART to issue an interrupt to the processor when the transmit data request interrupt enable is set high. Non-FIFO mode: 0 = No character transferred from the Transmit Holding register into the Transmit Shift register. 1 = A character has transferred from the Transmit Holding register into the Transmit Shift register. ***Note:*** Bit is reset to logic 0 with the loading of the Transmit Holding register by the processor. FIFO mode: 0 = When at least one byte is written to the transmit FIFO. When more than 16 characters are loaded into the FIFO, the excess characters are lost. 1 = Transmit FIFO is empty or the RESETTF bit in FCR, (Table 1041), has been set to 1.	1b	RO
04	BI	**Break Interrupt:** BI is set to a logic 1 when the received data input is held in the spacing (logic 0) state for longer than a full word transmission time (that is, the total time of Start bit + data bits + parity bit + stop bits).The BI is reset to a logic "0" when the processor reads the Line Status register. 0 = No Break signal has been received. 1 = Break signal occurred. In FIFO mode, only one character (equal to 00h), is loaded into the FIFO regardless of the length of the break condition. BI shows the break condition for the character at the top of the FIFO, not the most recently received character.	0	RO

Table 1058. Line Status Register (LSR) (Sheet 2 of 2)

Offset: Base + 05H	**Size:** 8 bit
Default Value: 60h	**Power Well:** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
03	FE	**Framing Error:** FE indicates that the received character did not have a valid stop bit. This bit is reset to a logic "0" when the processor reads the Line Status Register. 0 = No Framing error. 1 = Invalid stop bit has been detected. FE is set to a logic 1 when the bit following the last data bit or parity bit is detected as a logic 0 (spacing level). If the Line Control register had been set for two stop bit mode, the receiver does not check for a valid second stop bit. The FE indicator is reset when the processor reads the Line Status Register. The UART resynchronizes after a framing error. To do this it assumes that the framing error was due to the next start bit, so it samples this "start" bit twice and then takes in the "data". In FIFO mode FE shows a Framing error for the character at the top of the FIFO, not for the most recently received character.	0	RO
02	PE	**Parity Error:** PE indicates that the received data character does not have the correct even or odd parity, as selected by the even parity select bit. The PE is set to logic 1 upon detection of a parity error and is reset to a logic 0 when the processor reads the Line Status register. In FIFO mode, PE shows a parity error for the character at the top of the FIFO, not the most recently received character. 0 = No Parity error. 1 = Parity error has occurred.	0	RO
01	OE	**Overrun Error:** In non-FIFO mode, OE indicates that data in the receiver buffer register was not read by the processor before the next character was transferred into the receiver buffer register, thereby destroying the previous character. In FIFO mode, OE indicates that all 16 bytes of the FIFO are full and the most recently received byte has been discarded. The OE indicator is set to a logic "1" upon detection of an overrun condition and reset when the processor reads the Line Status Register. 0 = No data has been lost 1 = Received data has been lost.	0	RO
00	DR	**Data Ready:** DR is set to logic 1 when complete incoming character has been received and transferred into the Receiver Buffer Register (RBR) or the FIFO. In non-FIFO mode, DR is reset to 0 when the receive buffer is read. In FIFO mode, DR is reset to a logic 0 if the FIFO is empty (last character has been read from Receiver Buffer Register) or the RESETRF bit is set in FCR. 0 = No data has been received 1 = Data available in RBR or the FIFO.	0	RO

31.6.4.8 MCR – Modem Control Register

This 8-bit register controls the interface with the modem or data set (or a peripheral device emulating a modem).

Table 1059. MCR – Modem Control Register (Sheet 1 of 2)

Offset: Base + 04H	**Size:** 8 bit
Default Value: 00h	**Power Well:** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07:05	Reserved	Reserved	000b	
04	LOOP	**Loop back test mode:** This bit provides a local Loopback feature for diagnostic testing of the UART. When LOOP is set to a logic 1, the following occurs: The transmitter serial output is set to a logic 1 state. The OUT2# signal is forced to a logic 1 state. The receiver serial input is disconnected from the pin. The output of the Transmitter Shift register is "looped back" into the receiver shift register input. The four modem control inputs (CTS#, DSR#, DCD#, and RI#) are disconnected from the pins and the modem control output pins (RTS# and DTR#) are forced to their inactive state. • Coming out of the loopback test mode may result in unpredictable activation of the delta bits (bits 3:0) in the Modem Status Register (MSR). It is recommended that MSR be read once to clear the delta bits in the MSR. The lower four bits of the Modem Control register are connected to the upper four Modem Status register bits: • DTR = 1 forces DSR to a 1 • RTS = 1 forces CTS to a 1 • OUT1 = 1 forces RI to a 1 • OUT2= 1 forces DCD to a 1 In the diagnostic mode, data that is transmitted is immediately received. This feature allows the processor to verify the transmit and receive data paths of the UART. The transmit, receive and modem control interrupts are operational, except the modem control interrupts are activated by Control register bits, not the modem control inputs. A break signal can also be transferred from the transmitter section to the receiver section in loopback mode. 0 = Normal UART operation 1 = Test mode UART operation	0b	RW
03	OUT2	**OUT2# signal control:** This bit controls the OUT2# output. When the OUT2 bit is set, OUT2# is asserted low. When the OUT2 bit is cleared, OUT2# is deasserted (set high). Outside of the UART module, the OUT2# signal is used to connect the UART's interrupt output to the Interrupt Controller unit. 0 = OUT2# signal is 1, which disables the UART interrupt. 1 = OUT2# signal is 0.	0b	RW

Table 1059. MCR – Modem Control Register (Sheet 2 of 2)

Offset: Base + 04H	**Size:** 8 bit
Default Value: 00h	**Power Well:** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
02	OUT1	**Test bit:** This bit is used only in Loopback test mode. See (LOOP) Above.	0b	RW
01	RTS	**Request to Send:** This bit controls the Request to Send (RTS#) output pin. Bit 1 affects the RTS# output in a manner identical to that described below for the DTR bit. 0 = RTS# pin is 1 1 = RTS# pin is 0	0b	RW
00	DTR	**Data Terminal Ready:** This bit controls the Data Terminal Ready output. When bit 0 is set to a logic 1, the DTR# output is force to a logic 0. When bit 0 is reset to a logic 0, the DTR# output pin is forced to a logic 1. • The DTR# output of the UART may be applied to an EIA inverting line driver (such as the DS1488) to obtain the proper polarity input at the succeeding modem or data set. 0 = DTR# pin is 1 1 = DTR# pin is 0	0b	RW

31.6.4.9 MSR – Modem Status Register

This 8-bit register provides the current state of the control lines from the modem or data set (or a peripheral device emulating a modem) to the processor. In addition to this current state information, four bits of the Modem Status register provide change information. Bits 03:00 are set to a logic 1 when a control input from the Modem changes state. They are reset to a logic 0 when the processor reads the Modem Status register.

When bits 0, 1, 2, or 3 are set to logic 1, a Modem Status interrupt is generated if bit 3 of the Interrupt Enable Register is set.

Table 1060. MSR – Modem Status Register (Sheet 1 of 2)

Offset:	**Size:**
Default Value: 00h	**Power Well:** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07	DCD	**Data Carrier Detect:** This bit is the complement of the Data Carrier Detect (DCD#) input. This bit is equivalent to bit OUT2 of the Modem Control register if LOOP in the MCR is set to 1. 0 = DCD# pin is 1 1 = DCD# pin is 0	0b	RO
06	RI	**Ring Indicator:** This bit is the complement of the ring Indicator (RI#) input. This bit is equivalent to bit OUT1 of the Modem Control register if LOOP in the MCR is set to 1. 0 = RI# pin is 1 1 = RI# pin is 0	0b	RO
05	DSR	**Data Set Ready:** This bit is the complement of the Data Set Ready (DSR#) input. This bit is equivalent to bit DTR of the Modem Control register if LOOP in the MCR is set to 1. 0 = DSR# pin is 1 1 = DSR# pin is 0	0b	RO

Table 1060. MSR – Modem Status Register (Sheet 2 of 2)

Offset:	Size:
Default Value: 00h	Power Well: Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
04	CTS	**Clear to Send:** This bit is the complement of the Clear to Send (CTS#) input. This bit is equivalent to bit RTS of the Modem Control register if LOOP in the MCR is set to 1. 0 = CTS# pin is 1 1 = CTS# pin is 0	0b	RO
03	DDCD	Delta Data Carrier Detect: 0 = No change in DCD# pin since last read of MSR 1 = DCD# pin has changed state	0b	RO
02	TERI	Trailing Edge Ring Indicator: 0 = RI# pin has not changed from 0 to 1 since last read of MSR 1 = RI# pin has changed from 0 to 1	0b	RO
01	DDSR	Delta Data Set Ready: 0 = No change in DSR# pin since last read of MSR 1 = DSR# pin has changed state	0b	RO
00	DCTS	Delta Clear To Send: 0 = No change in CTS# pin since last read of MSR 1 = CTS# pin has changed state	0b	RO

31.6.4.10 SCR – Scratchpad Register

This 8-bit read/write register has no effect on the UART. It is intended as a scratchpad register for use by the programmer.

Table 1061. SCR – Scratchpad Register

Offset: Base + 07h	Size: 8 bit
Default Value: 00h	Power Well: Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07:00	SP[7:0]	No effect on UART functionality.	00h	RW

31.6.4.11 Programmable Baud Rate Generator

The UART contains a programmable Baud Rate Generator that is capable of taking the UART_CLK input and dividing it by any divisor from 1 to (2^{16} -1). The output frequency of the Baud Rate Generator is 16 times the baud rate. Two 8-bit latches store the divisor in a 16-bit binary format. These Divisor Latches must be loaded during initialization to ensure proper operation of the Baud Rate Generator. If both Divisor Latches are loaded with 0, the 16X output clock is stopped. Upon loading either of the Divisor latches, a 16-bit baud counter is immediately loaded. This prevents long counts on initial load. Access to the Divisor latch can be done with a word write.

Note: The UART_CLK is the SIW_CLK input divided by the prescalar set by the SIW Configuration Register (Offset 29h).

The baud rate of the data shifted in/out of the UART is given by:

Baud Rate = UART_CLK(MHz)/[16X Divisor]

For example, if UART_CLK is 14.7456MHz and the divisor is 96, the baud rate is 9600.

A Divisor value of 0 in the Divisor Latch Register is not allowed. The reset value of the divisor is 02.

Table 1062. DLL — Divisor Latch Register Low

Offset: *Base (DLAB=1)*		**Size:** *8 bit*		
Default Value: *02h*		**Power Well:** *Core*		
Lockable: *No*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:00	BR[7:0]	Low byte compare value to generate baud rate.	02h	RW

Table 1063. Divisor Register

Offset: *Base + 1 (DLAB=1)*		**Size:** *8 bit*		
Default Value: *00h*		**Power Well:** *Core*		
Lockable: *No*				
Bits	**Name**	**Description**	**Reset Value**	**Access**
07:00	BR[15:8]	High byte compare value to generate baud rate.	00h	RW

31.6.5 FIFO Operation

31.6.5.1 FIFO Interrupt Mode Operation

31.6.5.1.1 Receiver Interrupt

When the Receive FIFO and receiver interrupts are enabled (FCR[0]=1 and IER[0]=1), receiver interrupts occur as follows:

- The receive data available interrupt is invoked when the FIFO has reached its programmed trigger level. The interrupt is cleared when the FIFO drops below the programmed trigger level.
- The IIR receive data available indication also occurs when the FIFO trigger level is reached, and like the interrupt, the bits are cleared when the FIFO drops below the trigger level.
- The receiver line status interrupt (IIR = C6H), as before, has the highest priority. The receiver data available interrupt (IIR=C4H) is lower. The line status interrupt occurs only when the character at the top of the FIFO has errors.
- The data ready bit (DR in LSR register) is set to 1 as soon as a character is transferred from the shift register to the Receive FIFO. This bit is reset to 0 when the FIFO is empty.

31.6.5.1.2 Character Timeout Interrupt

When the receiver FIFO and receiver time out interrupt are enabled, a character timeout interrupt occurs when all of the following conditions exist:

- At least one character is in the FIFO.
- The last received character was longer than four continuous character times ago (if two stop bits are programmed the second one is included in this time delay).
- The most recent processor read of the FIFO was longer than four continuous character times ago.

- The receive FIFO trigger level is greater than one.

The maximum time between a received character and a timeout interrupt is 160 ms at 300 baud with a 12-bit receive character (i.e., one start, eight data, one parity, and two stop bits).

When a time out interrupt occurs, it is cleared and the timer is reset when the processor reads one character from the receiver FIFO. If a timeout interrupt has not occurred, the timeout timer is reset after a new character is received or after the processor reads the receiver FIFO.

31.6.5.1.3 Transmit Interrupt

When the transmitter FIFO and transmitter interrupt are enabled (FCR[0]=1, IER[1]=1), transmit interrupts occur as follows:

The transmitter holding register interrupt occurs when the transmit FIFO is empty; it is cleared as soon as the transmitter holder register is written to (1 to 16 characters may be written to the transmit FIFO while servicing this interrupt) or the IIR is read.

The transmitter FIFO empty indications are delayed one character time minus the last stop bit time whenever the following occurs: THRE = 1 and there have not been at least two bytes at the same time in the transmit FIFO since the last THRE = 1. The first transmitter interrupt after changing FCRO is immediate if it is enabled.

31.6.5.2 FIFO Polled Mode Operation

With the FIFOs enabled (TRFIFOE bit of FCR set to 1), setting IER[3:0] to all zeros puts the serial port in the FIFO polled mode of operation. Since the receiver and the transmitter are controlled separately, either one or both can be in the polled mode of operation. In this mode, software checks receiver and transmitter status via the LSR. As stated in the register description:

- LSR[0] is set as long as there is one byte in the receiver FIFO.
- LSR[1] through LSR[4] specify which error(s) has occurred for the character at the top of the FIFO. Character error status is handled the same way as interrupt mode. The IIR is not affected since IER[2] = 0.
- LSR[5] indicates when the transmitter FIFO needs data.
- LSR[6] indicates that both the transmitter FIFO and shift register are empty.
- LSR[7] indicates whether there are any errors in the receiver FIFO.

31.7 Logical Device 6: Watchdog Timer

31.7.1 Overview

This device is a Watchdog timer that provides a resolution that ranges from 1 µs to 1050 s. The timer uses a 35-bit down-counter.

The counter is loaded with the value from the 1st Preload register. The timer is then enabled and it starts counting down. The time at which the WDT first starts counting down is called the first stage. If the host fails to reload the WDT before the 35-bit down counter reaches zero the WDT generates an internal interrupt.

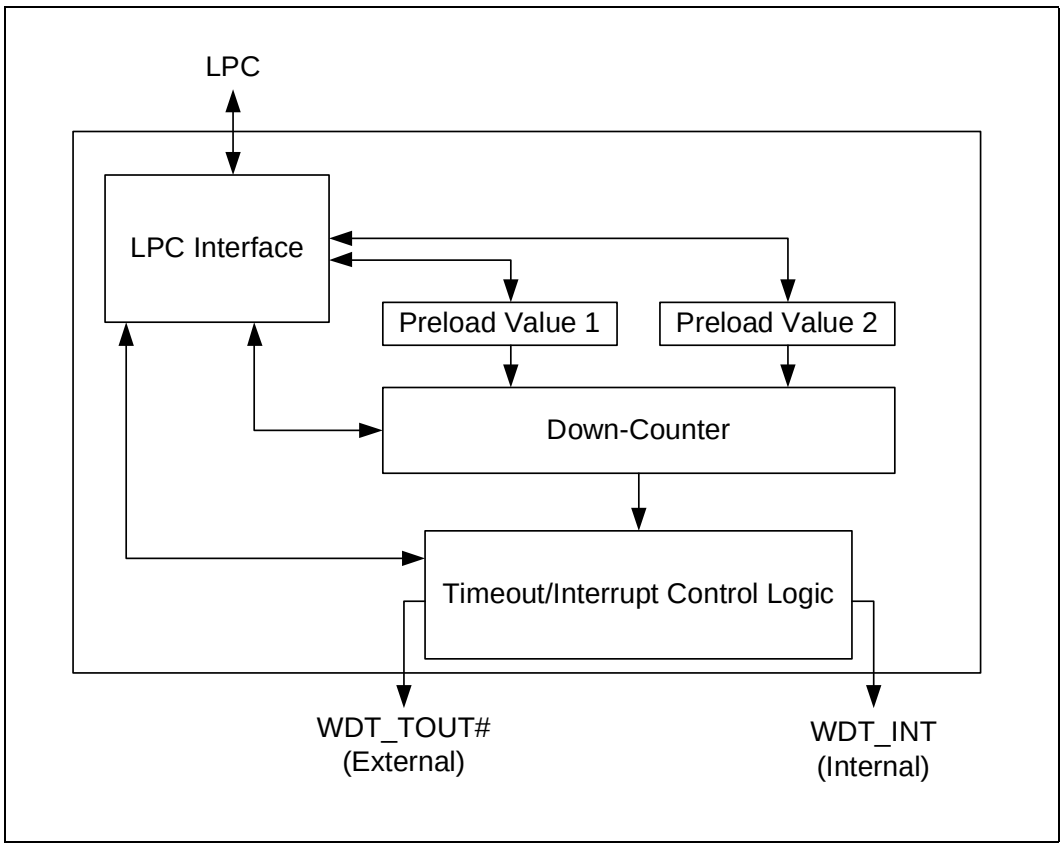
After the interrupt is generated the WDT loads the value from the 2nd Preload register into the WDT's 35-bit Down-Counter and starts counting down. The WDT is now in the second stage. If the host still fails to reload the WDT before the second timeout. The WDT drives the WDT_TOUT# pin low and sets the timeout bit (WDT_TIMEOUT). This bit

indicates that the System has become unstable. The WDT_TOUT# pin is held low until the system is Reset or the WDT times out again (Depends on TOUT_CNF). The process of reloading the WDT involves the following sequence of writes:

1. Write “80” to offset BAR1 + 0Ch
2. Write “86” to offset BAR1 + 0Ch
3. Write ‘1’ to WDT_RELOAD in Reload Register.

The same process is used for setting the values in the preload registers. The only difference exists in step 3. Instead of writing a ‘1’ to the WDT_RELOAD, you write the desired preload value into the corresponding Preload register. This value is not loaded into the 35-bit down counter until the next time the WDT reenters the stage. For example, if Preload Value 2 is changed, it is not loaded into the 35-bit down counter until the next time the WDT enters the second stage.

Figure 99. WDT Block Diagram



31.7.2 I/O Registers

All registers not mentioned are reserved.

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 1064. I/O Register Summary Table

Offset	Register	Default	Type
Base+00h	Preload Value 1 Register 0	FFh	RW
Base+01h	Preload Value 1 Register 1	FFh	RW
Base+02h	Preload Value 1 Register 2	0Fh	RO, RW
Base+04h	Preload Value 2 Register 0	FFh	RW
Base+05h	Preload Value 2 Register 1	FFh	RW
Base+06h	Preload Value 2 Register 2	0Fh	RO, RW
Base+08h	General Interrupt Status Register	00h	RWC, RO
Base+0ch	Reload Register 0	00h	W
Base+0dh	Reload Register 1	00h	RO, RW
Base+10h	WDT Configuration Register 0	00h	RO, RW
Base+18h	WDT Lock Register	00h	RO, RW1, RW

Any reads to this offset return an indeterminate value.

31.7.2.1 Preload Value 1 Register 0

Table 1065. Preload Value 1 Register 0

Offset: Base + 00h ***Size:*** 8 bit
Default Value: FFh ***Power Well:*** Core
Lockable: No

Bits	Name	Description	Reset Value	Access
07:00	PLOAD1_7_0	**Preload_Value_1 [7:0]:** This register is used to hold the bits 0 through 7 of the preload value 1 for the WDT Timer. The Value in the Preload Register is automatically transferred into the 35-bit down counter every time the WDT enters the first stage. The value loaded into the preload register needs to be one less than the intended period. This is because the timer makes use of zero-based counting (i.e. zero is counted as part of the decrement). Refer to Section 31.7.3.2 for details on how to change the value of this register.	FFh	RW

31.7.2.2 Preload Value 1 Register 1

Table 1066. Preload Value 1 Register 1

Offset: Base + 01h	***Size:*** 8 bit
Default Value: FFh	***Power Well:*** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07:00	PLOAD1_15_8	**Preload_Value_1 [15:8]:** This register is used to hold the bits 8 through 15 of the preload value 1 for the WDT Timer. The Value in the Preload Register is automatically transferred into the 35-bit down counter every time the WDT enters the first stage. The value loaded into the preload register needs to be one less than the intended period. This is because the timer makes use of zero-based counting (i.e. zero is counted as part of the decrement). Refer to Section 31.7.3.2 for details on how to change the value of this register.	FF	RW

31.7.2.3 Preload Value 1 Register 2

Table 1067. Preload Value 1 Register 2

Offset: Base + 02h	***Size:*** 8 bit
Default Value: 0Fh	***Power Well:*** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	0h	
03:00	PLOAD_19_16	**Preload_Value_1 [19:16]:** This register is used to hold the bits 16 through 19 of the preload value 1 for the WDT Timer. The Value in the Preload Register is automatically transferred into the 35-bit down counter every time the WDT enters the first stage. The value loaded into the preload register needs to be one less than the intended period. This is because the timer makes use of zero-based counting (i.e. zero is counted as part of the decrement). Refer to Section 31.7.3.2 for details on how to change the value of this register.	Fh	RW

31.7.2.4 Preload Value 2 Register 0

Table 1068. Preload Value 2 Register 0

Offset: Base + 04h	***Size:*** 8 bit
Default Value: FFh	***Power Well:*** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07:00	PLOAD2_7_0	**Preload_Value_2 [7:0]:** This register is used to hold the bits 0 through 7 of the preload value2 for the WDT Timer. The Value in the Preload Register is automatically transferred into the 35-bit down counter every time the WDT enters the second stage. The value loaded into the preload register needs to be one less than the intended period. This is because the timer makes use of zero-based counting (i.e. zero is counted as part of the decrement). Refer to Section 31.7.3.2 for details on how to change the value of this register.	FF	RW

31.7.2.5 Preload Value 2 Register 1

Table 1069. Preload Value 2 Register 1

Offset: Base + 05h	***Size:*** 8 bit
Default Value: FFh	***Power Well:*** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07:00	PLOAD2_15_8	**Preload_Value_2 [15:8]:** This register is used to hold the bits 8 through 15 of the preload value2 for the WDT Timer. The Value in the Preload Register is automatically transferred into the 35-bit down counter every time the WDT enters the second stage. The value loaded into the preload register needs to be one less than the intended period. This is because the timer makes use of zero-based counting (i.e. zero is counted as part of the decrement). Refer to Section 31.7.3.2 for details on how to change the value of this register.	FF	RW

31.7.2.6 Preload Value 2 Register 2

Table 1070. Preload Value 2 Register 2

Offset: Base + 06h	***Size:*** 8 bit
Default Value: 0Fh	***Power Well:*** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07:04	Reserved	Reserved	0	
03:00	PLOAD2_19_16	**Preload_Value_2 [19:16]:** This register is used to hold the bits 16 through 19 of the preload value2 for the WDT Timer. The Value in the Preload Register is automatically transferred into the 35-bit down counter every time the WDT enters the second stage. The value loaded into the preload register needs to be one less than the intended period. This is because the timer makes use of zero-based counting (i.e. zero is counted as part of the decrement). Refer to Section 31.7.3.2 for details on how to change the value of this register.	F	RW

31.7.2.7 General Interrupt Status Register

Table 1071. General Interrupt Status Register (Sheet 1 of 2)

Offset: Base + 08h	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07:03	Reserved	Reserved	00h	

Table 1071. General Interrupt Status Register (Sheet 2 of 2)

Offset: Base + 08h	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
02	SMIACT	**Watchdog Timer SMI Interrupt Active (1st Stage):** This bit is set when the first Stage of the 35-bit Down Counter Reaches zero. An SMI interrupt is generated if WDT_INT_TYPE is configured to do so (See WDT Configuration Register). This is a sticky bit and is only cleared by writing a '1'. This SMI interrupt will be routed to GPIO6 so that BIOS can use the existing SMM handler to service this interrupt. 0 = No Interrupt 1 = Interrupt Active ***Note:*** This bit is not set in free running mode. Also note that to route the SMI interrupt to GPI6, the GPI_ROUT[13:12] register in D31:F0:B8h must be set to "01" to generate the WDT SMI interrupt and GPIO6 cannot be used as a general purpose input pin.	0	RWC
01	NMIACT	**Watchdog Timer NMI Interrupt Active (1st Stage):** This bit is set when the first Stage of the 35-bit Down Counter Reaches zero. An NMI interrupt is generated if WDT_INT_TYPE is configured to do so (See WDT Configuration Register). This is a sticky bit and is only cleared by writing a '1'. 0 = No Interrupt 1 = Interrupt Active ***Note:*** This bit is not set in free running mode.	0	RWC
00	SERIRQACT	**Watchdog Timer SERIRQ Interrupt Active (1st Stage):** This bit is set when the first Stage of the 35-bit Down Counter Reaches zero. An SERIRQ interrupt is generated if WDT_INT_TYPE is configured to do so (See WDT Configuration Register). This is a sticky bit and is only cleared by writing a '1'. 0 = No Interrupt 1 = Interrupt Active ***Note:*** This bit is not set in free running mode.	0	RWC

31.7.2.8 Reload Register 0

Table 1072. Reload Register 0

Offset: Base + 0Ch	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07:00	Reserved	Reserved. Must be programmed to 0.	00h	

31.7.2.9 Reload Register 1

Table 1073. Reload Register 1

Offset: Base + 0Dh	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07:02	Reserved	Reserved	00	
01	TOUT	**WDT_TIMEOUT:** This bit is located in the RTC Well and it's value is not lost if the host resets the system. It is set to '1' if the host fails to reset the WDT before the 35-bit Down-Counter reaches zero for the second time in a row. This bit is cleared by performing the Register Unlocking Sequence followed by a '1' to this bit. 0 = Normal (Default) 1 = System has become unstable. ***Note:*** In free running mode this bit is set every time the down counter reaches zero.	0	RW
00	RELOAD	**WDT_RELOAD:** To prevent a timeout the host must perform the Register Unlocking Sequence followed by a '1' to this bit. Refer to Section 31.7.3.2 for details on how to change the value of this register.	0	RW

31.7.2.10 WDT Configuration Register

Table 1074. WDT Configuration Register (Sheet 1 of 2)

Offset: Base + 10h	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
07:06	Reserved	Reserved	00	
05	WDT_TOUT_EN	**WDT Timeout Output Enable:** This bit indicates whether or not the WDT toggles the external WDT_TOUT# pin if the WDT times out. 0 = Enabled (Default) 1 = Disabled	0	RW

Table 1074. WDT Configuration Register (Sheet 2 of 2)

Offset: Base + 10h	***Size:*** 8 bit
Default Value: 00h	***Power Well:*** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
04:03	Reserved	Reserved	00	
02	WDT_PRE_SEL	**WDT Prescaler Select:** The WDT provides two options for prescaling the main Down Counter. The preload values are loaded into the main down counter right justified. The prescaler adjusts the starting point of the 35-bit down counter. 0 = The 20-bit Preload Value is loaded into bits 34:15 of the main down counter. The resulting timer clock is the PCI Clock (33 MHz) divided by 2^{15}. The approximate clock generated is 1 KHz, (1 ms to 1050 sec). (Default) 1 = The 20-bit Preload Value is loaded into bits 24:05 of the main down counter. The resulting timer clock is the PCI Clock (33 MHz) divided by 2^{5}. The approximate clock generated is 1 MHz, (1 µs to 1sec)	0	RW
01:00	WDT_INT_TYPE	**WDT_INT_TYPE:** The WDT timer supports programmable routing of interrupts. The set of bits allows the user to choose the type of interrupt desired if the WDT reached the end of the first stage without being reset. The interrupt status is reported in the WDT General Interrupt Status register. 00 SERIRQ (Default) 01 NMI 10 SMI 11 Disabled ***Note:*** SERIRQ is Active Low	00	RW

31.7.2.11 WDT Lock Register

Table 1075. WDT Lock Register (Sheet 1 of 2)

Offset: *Base + 18h*	***Size:*** 8 bit
Default Value: *00h*	***Power Well:*** Core
Lockable: *No*	

Bits	Name	Description	Reset Value	Access
07:03	Reserved	Reserved	0	
02	WDT_TOUT_CNF	**WDT Timeout Configuration:** This register is used to choose the functionality of the timer. 0 = Watchdog Timer Mode: When enabled (i.e. WDT_ENABLE goes from '0' to '1') the timer reloads Preload Value 1 and start decrementing. (Default) Upon reaching the second stage timeout the WDT_TOUT# is driven low once and does not change again until Power is cycled or a hard reset occurs. 1 = Free Running Mode: WDT_TOUT# changes from previous state when the next timeout occurs. The timer ignores the first stage. The timer only uses Preload Value 2. In this mode the timer is restarted whenever WDT_ENABLE goes from a 0 to a 1. This means that the timer reloads Preload Value 2 and start decrementing every time it is enabled. In free running mode it is not necessary to reload the timer as it is done automatically every time the descrementer reaches zero.	0	RW

Table 1075. WDT Lock Register (Sheet 2 of 2)

Offset: Base + 18h	**Size:** 8 bit
Default Value: 00h	**Power Well:** Core
Lockable: No	

Bits	Name	Description	Reset Value	Access
01	WDT_ENABLE	**Watchdog Timer Enable:** The following bit enables or disables the WDT. 0 = Disabled (Default) 1 = Enabled ***Note:*** This bit cannot be modified if WDT_LOCK has been set. ***Note:*** In free-running mode Preload Value 2 is reloaded into the down counter every time WDT_ENABLE goes from '0' to '1'. ***Note:*** In WDT mode Preload Value 1 is reloaded every time WDT_ENABLE goes from '0' to '1' or the WDT_RELOAD bit is written using the proper sequence of writes (See Register Unlocking Sequence). When the WDT second stage timeout occurs, a reset must happen. ***Note:*** Software must guarantee that a timeout is not about to occur before disabling the timer. A reload sequence is suggested.	0	RW
00	WDT_LOCK	**Watchdog Timer Lock:** Setting this bit locks the values of this register until a hard-reset occurs or power is cycled. 0 = Unlocked (Default) 1 = Locked ***Note:*** Writing a "0" has no effect on this bit. Write is only allowed from "0" to "1" once. It cannot be changed until either power is cycled or a hard-reset occurs.	0	RW1

31.7.3 Theory Of Operation

31.7.3.1 RTC Well and WDT_TOUT# Functionality

The WDT_TIMEOUT bit is set to a '1' when the WDT 35-bit down counter reaches zero for the second time in a row. Then the WDT_TOUT# pin is toggled LOW by the WDT from the IICH. The board designer must attach the WDT_TOUT# to the appropriate external signal. If WDT_TOUT_CNF is a '1' the WDT toggles WDT_TOUT# again the next time a time out occurs. Otherwise WDT_TOUT# is driven low until the system is reset or power is cycled.

31.7.3.2 Register Unlocking Sequence

The register unlocking sequence is necessary whenever writing to the RELOAD register or either PRELOAD_VALUE registers. The host must write a sequence of two writes to offset BAR1 + 0Ch before attempting to write to either the WDT_RELOAD and WDT_TIMEOUT bits of the RELOAD register or the PRELOAD_VALUE registers. The first writes are "80" and "86" (in that order) to offset BAR1 + 0Ch. The next write is to the proper memory mapped register (e.g., RELOAD, PRELOAD_VALUE_1, PRELOAD_VALUE_2). Any deviation from the sequence (writes to memory-mapped registers) causes the host to have to restart the sequence.

When performing register unlocking, software must issue the cycles using byte access only. Otherwise the unlocking sequence will not work properly.

The following is an example of how to prevent a timeout:

1. Write "80" to offset BAR1 + 0Ch

2. Write “86” to offset BAR1 + 0Ch
3. Write a ‘1’ to RELOAD [8] (WDT_RELOAD) of the Reload Register

Note: Any subsequent writes require that this sequence be performed again.

31.7.3.3 Reload Sequence

To keep the timer from causing an interrupt or driving WDT_TOUT#, the timer must be updated periodically. Other timers refer to “updating the timer” as “kicking the timer”. The frequency of updates required is dependent on the value of the Preload values. To update the timer the Register Unlocking Sequence must be performed followed by writing a ‘1’ to bit 8 at offset BAR1 + 0Ch within the watchdog timer memory mapped space. This sequence of events is referred to as the “Reload Sequence”.

31.7.3.4 Low Power State

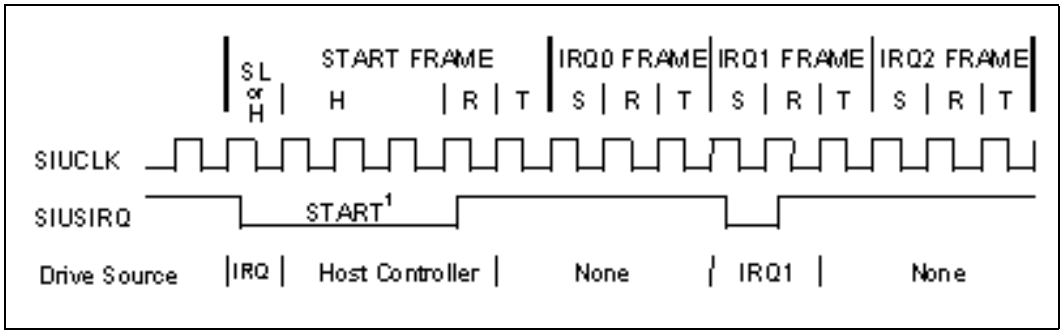
The Watchdog Timer does not operate when PCICLK is stopped.

31.8 Serial IRQ

The SIW supports the serial interrupt to transmit interrupt information to the host system. The serial interrupt scheme adheres to the *Serial IRQ Specification*.

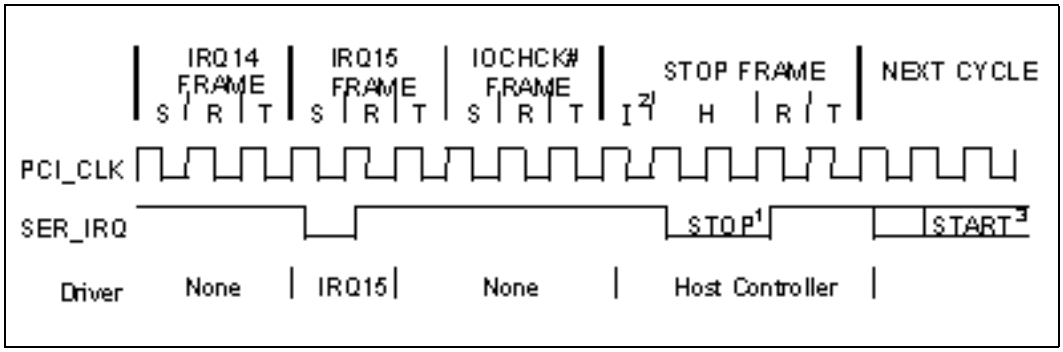
31.8.1 Timing Diagrams For SIW_SERIRQ Cycle

Figure 100. Start Frame Timing with Source Sampled a Low Pulse on IRQ1



Notes:
1. H=Host Control; R=Recovery; T=Turn-Around; SL=Slave Control; S=Sample
2. Start Frame pulse can be 4-8 clocks wide depending on the location of the device in the PCI bridge hierarchy in a synchronous bridge design.

Figure 101. Stop Frame Timing with Host Using Quiet Mode Sampling Period



Notes:
1. H=Host Control; R=Recovery; T=Turn-Around; S=Sample; I=Idle
2. Stop pulse is two clocks wide for Quiet mode, three clocks wide for Continuous mode.
3. There may be none, one or more Idle states during the Stop Frame.
4. The next SIW_SERIRQ cycle's Start Frame pulse may or may not start immediately after the turn-around clock of the Stop Frame.

31.8.1.1 SIW_SERIRQ Cycle Control

There are two modes of operation for the SIW_SERIRQ Start Frame.

1. **Quiet (Active) Mode**: Any device may initiate a Start Frame by driving the SIW_SERIRQ low for one clock, while the SIW_SERIRQ is Idle. After driving low for one clock the SIW_SERIRQ is immediately tri-stated without at any time driving high. A Start Frame may not be initiated while the SIW_SERIRQ is Active. The SIW_SERIRQ is Idle between Stop and Start Frames. The SIW_SERIRQ is Active between Start and Stop Frames. This mode of operation allows the SIW_SERIRQ to be Idle when there are no IRQ/Data transitions which should be most of the time.

   Once a Start Frame has been initiated the Host Controller takes over driving the SIW_SERIRQ low in the next clock and continues driving the SIW_SERIRQ low for a

programmable period of three to seven clocks. This makes a total low pulse width of four to eight clocks. Finally, the Host Controller drives the SIW_SERIRQ back high for one clock, then tri-state.

Any SIW_SERIRQ Device (i.e., The SIU and WDT) which detects any transition on an IRQ/Data line for which it is responsible must initiate a Start Frame in order to update the Host Controller unless the SIW_SERIRQ is already in an SIW_SERIRQ Cycle and the IRQ/Data transition can be delivered in that SIW_SERIRQ Cycle.

2. **Continuous (Idle) Mode**: Only the Host controller can initiate a Start Frame to update IRQ/ Data line information. All other SIW_SERIRQ agents become passive and may not initiate a Start Frame. SIW_SERIRQ is driven low for four to eight clocks by Host Controller. This mode has two functions. It can be used to stop or idle the SIW_SERIRQ or the Host Controller can operate SIW_SERIRQ in a continuous mode by initiating a Start Frame at the end of every Stop Frame.

   An SIW_SERIRQ mode transition can only occur during the Stop Frame. Upon reset, SIW_SERIRQ bus is defaulted to Continuous mode, therefore only the Host controller can initiate the first Start Frame. Slaves must continuously sample the Stop Frames pulse width to determine the next SIW_SERIRQ Cycle's mode.

31.8.1.2 SIW_SERIRQ Data Frame

Once a Start Frame has been initiated, the SIW watches for the rising edge of the Start Pulse and start counting IRQ/Data Frames from there. Each IRQ/Data Frame is three clocks: Sample phase, Recovery phase, and Turn-around phase. During the Sample phase the SIW drives the SIW_SERIRQ low, if and only if, its last detected IRQ/Data value was low. If its detected IRQ/Data value is high, SIW_SERIRQ is left tri-stated. During the Recovery phase the SIW drives the SIW_SERIRQ high, if and only if, it had driven the SIW_SERIRQ low during the previous Sample Phase. During the Turn-around Phase the SIW tri-states the SIU_SERIRQ. The SIW drives the SIW_SERIRQ line low at the appropriate sample point if its associated IRQ/Data line is low, regardless of which device initiated the Start Frame.

The Sample Phase for each IRQ/Data follows the low to high transition of the Start Frame pulse by a number of clocks equal to the IRQ/Data Frame times three, minus one. (e.g., The IRQ5 Sample clock is the sixth IRQ/Data Frame, (6 x 3) - 1 = 17th clock after the rising edge of the Start Pulse).

Table 1076. SIW_SERIRQ Sampling Periods (Sheet 1 of 2)

SIW_SERIRQ PERIOD	SIGNAL SAMPLED	# OF CLOCKS PAST START
1	Not Used	2
2	IRQ1	5
3	IRQ2	8
4	IRQ3	11
5	IRQ4	14
6	IRQ5	17
7	IRQ6	20
8	IRQ7	23
9	IRQ8	26
10	IRQ9	29
11	IRQ10	32
12	IRQ11	35
13	IRQ12	38

Table 1076. SIW_SERIRQ Sampling Periods (Sheet 2 of 2)

SIW_SERIRQ PERIOD	SIGNAL SAMPLED	# OF CLOCKS PAST START
14	IRQ13	41
15	IRQ14	44
16	IRQ15	47

SIW_SERIRQ Period 14 is used to transfer IRQ13. Logical devices 4 (Serial Port 1), 5 (Serial Port 2) and 6 (WDT) shall have IRQ13 as a choice for their primary interrupt.

31.8.1.3 Stop Cycle Control

Once all IRQ/Data Frames have completed the Host Controller terminates SIW_SERIRQ activity by initiating a Stop Frame. Only the Host Controller can initiate the Stop Frame. A Stop Frame is indicated when the SIW_SERIRQ is low for two or three clocks. If the Stop Frame's low time is two clocks then the next SIW_SERIRQ Cycle's sampled mode is the Quiet mode; and any SIW_SERIRQ device may initiate a Start Frame in the second clock or more after the rising edge of the Stop Frame's pulse. If the Stop Frame's low time is three clocks then the next SIW_SERIRQ Cycle's sampled mode is the Continuous mode; and only the Host Controller may initiate a Start Frame in the second clock or more after the rising edge of the Stop Frame's pulse.

31.8.1.4 Latency

Latency for IRQ/Data updates over the SIW_SERIRQ bus in bridge-less systems with the minimum Host supported IRQ/Data Frames of seventeen, ranges up to 96 clocks (2.88 µs with a 33 MHz PCI Bus). If one or more PCI to PCI Bridge is added to a system, the latency for IRQ/ Data updates from the secondary or tertiary buses are a few clocks longer for synchronous buses, and approximately double for asynchronous buses.

31.8.1.5 EOI / ISR Read Latency

Any serialized IRQ scheme has a potential implementation issue related to IRQ latency. IRQ latency could cause an EOI or ISR Read to precede an IRQ transition that it should have followed. This could cause a system fault. The host interrupt controller is responsible for ensuring that these latency issues are mitigated. The recommended solution is to delay EOIs and ISR Reads to the interrupt controller by the same amount as the SIW_SERIRQ Cycle latency in order to ensure that these events do not occur out of order.

31.8.1.6 Reset and Initialization

The SIW_SERIRQ bus uses SIW_LRESET# as its reset signal. The SIW_SERIRQ pin is tri-stated by all agents while SIW_LRESET# is active. With reset, SIW_SERIRQ Slaves are put into the (continuous) IDLE mode. The Host Controller is responsible for starting the initial SIW_SERIRQ Cycle to collect system's IRQ/Data default values. The system then follows with the Continuous/ Quiet mode protocol (Stop Frame pulse width) for subsequent SIW_SERIRQ Cycles. It is Host Controller's responsibility to provide the default values to the Interrupt controller and other system logic before the first SIW_SERIRQ Cycle is performed. For SIW_SERIRQ system suspend, insertion, or removal application, the Host controller must be programmed into Continuous (IDLE) mode first. This is to guarantee SIW_SERIRQ bus is in IDLE state before the system configuration changes.

31.9 Configuration

The configuration of the SIW is very flexible and is based on the configuration architecture implemented in typical Plug-and-Play components. The SIW is designed for motherboard applications in which the resources required by their components are known. With its flexible resource allocation architecture, the SIW allows the BIOS to assign resources at POST.

31.9.1 Configuration Port Address

The SIW configuration port addresses for INDEX and DATA are controlled by the strap pin SIU1_DTR# during reset. When SIU1_DTR# is driven to ‘1’ or left floating during reset SIW configuration port addresses are fixed at 4Eh/4Fh. When SIU1_DTR# is driven to ‘0’ during reset SIW configuration port addresses are fixed at 20Eh/20Fh.

31.9.2 Primary Configuration Address Decoder

After a PCI Reset (SIW_LRESET# pin asserted) or Power On Reset the SIW is in the Run Mode with the UARTs and Watchdog timer disabled. They may be configured through two standard Configuration I/O Ports (INDEX and DATA) by placing the SIW into Configuration Mode.

The BIOS uses these configuration ports to initialize the logical devices at POST. The INDEX and DATA ports are only valid when the SIW is in Configuration Mode.

The INDEX and DATA ports are effective only when the chip is in the Configuration State. When the SIW is not in the Configuration State, reads return FFh and write data is ignored.

31.9.2.1 Entering the Configuration State

The device enters the Configuration State by the following contiguous sequence:

1. Write 80H to Configuration Index Port.
2. Write 86H to Configuration Index Port.

31.9.2.2 Exiting the Configuration State

The device exits the Configuration State by the following contiguous sequence:

1. Write 68H to Configuration Index Port
2. Write 08H to Configuration Index Port

31.9.2.3 Configuration Sequence

To program the configuration registers, the following sequence must be followed.

1. Enter Configuration Mode
2. Configure the Configuration Registers
3. Exit Configuration Mode.

31.9.2.4 Configuration Mode

The system sets the logical device information and activates desired logical devices through the INDEX and DATA ports. In configuration mode, the INDEX PORT is located at the CONFIG PORT address and the DATA PORT is at INDEX PORT address + 1.

The desired configuration registers are accessed in two steps:

1. Write the index of the Logical Device Number Configuration Register (i.e., 07) to the INDEX PORT and then write the number of the desired logical device to the DATA PORT.
2. Write the address of the desired configuration register within the logical device to the INDEX PORT and then write or read the configuration register through the DATA PORT.

Note: If accessing the Global Configuration Registers, step (1) is not required. The chip returns to the RUN State.

Note: Only two states are defined (Run and Configuration). In the Run State the chip is always ready to enter the Configuration State.

31.9.3 SIW Configuration Register Summary

The default values are defined with an h for hex, a b for binary, or 00 for zero. If there is not a letter following the default value, assume it is a binary number.

Warning: Address locations that are not listed are considered reserved register locations. Reads to reserved registers may return non-zero values. Writes to reserved locations may cause system failure.

Note: Reserved bits are Read Only.

Table 1077. Configuration Register Summary (Sheet 1 of 2)

Global Configuration Registers			
Index	**Type**	**Default**	**Configuration Register**
07h	RW	00h	Logical Device Number
20h	R	00h	Device ID
21h	R	01h	Device Rev
28h	RW	01h	SIW I/F (wait states)
29h	RW	02h	SIRQ Configuration
2Eh	RW	00h	Test Mode Configuration Register
Logical Device 4 Registers (Serial Port 1)			
30h	RW	00h	Enable
60h	RW	00h	Base I/O Address MSB
61h	RW	00h	Base I/O Address LSB
70h	RW	00h	Primary Interrupt Select
74h	RW	04h	RSVD
75h	RW	04h	RSVD
F0h	RW	00h	RSVD
Logical Device 5 Registers (Serial Port 2)			
30h	RW	00h	Enable
60h	RW	00h	Base I/O Address MSB
61h	RW	00h	Base I/O Address LSB
70h	RW	00h	Primary Interrupt Select
74h	RW	04h	RSVD

Table 1077. Configuration Register Summary (Sheet 2 of 2)

75h	RW	04h	RSVD
F0h	RW	00h	RSVD
Logical Device 6 Registers (Watchdog Timer)			
30h	RW	00h	Enable
60h	RW	00h	Base I/O Address MSB
61h	RW	00h	Base I/O Address LSB
70h	RW	00h	Primary Interrupt Select

31.9.3.1 Global Control/Configuration Registers [00h - 2Fh]

The chip-level (global) registers lie in the address range [00h-2Fh]. The design MUST use all eight bits of the ADDRESS Port for register selection. All unimplemented registers and bits ignore writes and return zero when read.

The INDEX PORT is used to select a configuration register in the chip. The DATA PORT is then used to access the selected register. These registers are accessible only in the Configuration Mode.

Register	Address (Type)	Description
Logical Device #` Default = 00h	07h (RW)	**Logical Device Select**: A write to this register selects the current logical device. This allows access to the control and configuration registers for each logical device.
Device ID Default = 00h	20h (R)	**Device ID:** A read only register which provides the Device ID.
Device Rev Default = 01h	21h (R)	**Device Rev:** A read only register which provides device revision information.
SIW Interface Default = 01h	28h	Bit 7:1 RSVD = 0 Bit 0 LPC bus wait states 0 = Not Supported 1 = Long wait states (sync 6)

Register	Address (Type)	Description
SIW Configuration Default = 02h	29h (RW - bit 0, 2, 3) (R - bit 1)	Bit 0 SIRQ enable 1 =enabled; enabled logical devices participate in interrupt generation 0 =disabled; serial interrupts disabled Bit 1 IRQ mode (READ ONLY, WRITES IGNORED) 1 =Continuous mode 0 =Quiet mode Bit 3:2 UART_CLK predivide 00 = divide by 1 01 = divide by 8 10 = divide by 26 11 = reserved Bit 7:4 RSVD = 0
SIW Monitor Port Control Register Default = 00h	2D (RO - bit 7) (RW - bits 6:0)	Bit 0 UART1 Monitor Port Enable (UART1_MONPORTEN): Setting this bit enables the monitor port for UART1. This signal turns on all 8 UART2 monitor ports. Note: wdt_monporten, uart2_monporten and uart1_monporten must be set in a mutually exclusive manner i.e., only one monitor port enable must be set at one time. Bit 1 UART2 Monitor Port Enable (UART2_MONPORTEN): Setting this bit enables the monitor port for UART2. This signal turns on all 8 UART1 monitor ports. Note: wdt_monporten, uart2_monporten and uart1_monporten must be set in a mutually exclusive manner i.e., only one monitor port enable must be set at one time. Bit 2 WDT Monitor Port Enable (WDT_MONPORTEN): Setting this bit enables the monitor port for WDT. This signal turns on all 16 SIW monitor ports. Note: wdt_monporten, uart2_monporten and uart1_monporten must be set in a mutually exclusive manner i.e., only one monitor port enable must be set at one time. Bit 6:3 Monitor Port Slot Select[3:0] (MONPORTSEL): These bits select which Monitor Port Slot is enabled. These bits are used to select up to 16 slots within each UART1 and UART2 source group. Note: WDT has only 1 slot. So, port slot selection is not required. Bit 7 RSVD = 0
Default = 00h	2Eh	Reserved

31.9.3.2 Logical Device Configuration Registers [30h — FFh]

Used to access the registers that are assigned to each logical unit. This chip supports three logical units and has three sets of logical device registers. The three logical devices are UART1, UART2 and Watchdog Timer. A separate set (bank) of control and configuration registers exists for each logical device and is selected with the Logical Device # Register.

The INDEX PORT is used to select a specific logical device register. These registers are then accessed through the DATA PORT.

The Logical Device registers are accessible only when the device is in the Configuration State. The logical register addresses are shown in the tables below.

Table 1078. Logical Device 4 (Serial Port 1)

Logical Device Register	Address	Description

Enable
Default = 00h
	30h
	(RW)
		Bits[7:1] Reserved, set to zero.
		Bit[0]
		1 =enable the logical device currently selected through the Logical Device # register.
		0 =Logical device currently selected is inactive

I/O Base Address Default = 00h	60h (RW) 61h (Bits 7:3 RW Bits 2:0 RO)	Registers 60h (MSB) and 61h (LSB) set the base address for the device. Note: Decode is on 8 Byte boundaries Comm Decode Ranges 3F8 - 3FF (COM 1) 2F8 - 2FF (COM 2) 220 - 227 228 - 22F 238 - 23F 2E8 - 2EF (COM 4) 338 - 33F 3E8 - 3EF (COM 3)
Primary Interrupt Select Default = 00h	70h (RW)	Bits[3:0] selects which interrupt level is used for the primary Interrupt. 00= no interrupt selected 01= IRQ1 02= IRQ2 03= IRQ3 04= IRQ4 05= IRQ5 06= IRQ6 07= IRQ7 08= IRQ8 09= IRQ9 0A= IRQ10 0B= IRQ11 0C= IRQ12 0D= IRQ13 0E= IRQ14 0F= IRQ15 Bits[7:4] Reserved Note: An Interrupt is activated by enabling this device (offset 30h), setting this register to a non-zero value and setting any combination of bits 0-4 in the corresponding UART IER and the occurrence of the corresponding UART event (i.e. Modem Status Change, Receiver Line Error Condition, Transmit Data Request, Receiver Data Available or Receiver Time Out) and setting the OUT2 bit in the MCR.
Reserved Default = 04h	74h	Bit 7:0 - Reserved
Reserved Default = 04h	75h	Bit 7:0 - Reserved
Reserved Default = 00h	F0h	Bit 7:0 - Reserved

Table 1079. Logical Device 5 (Serial Port 2)

Logical Device Register	Address	Description

Enable
Default = 00h
	30h
	(RW)
		Bits[7:1] Reserved, set to zero.
		Bit[0]
		1 =enable the logical device currently selected through the Logical Device # register.
		0 =Logical device currently selected is inactive

I/O Base Address Default = 00h	60h (RW) 61h (Bits 7:3 RW Bits 2:0 RO)	Registers 60h (MSB) and 61h (LSB) set the base address for the device. Note: Decode is on 8 Byte boundaries Comm Decode Ranges 3F8 - 3FF (COM 1) 2F8 - 2FF (COM 2) 220 - 227 228 - 22F 238 - 23F 2E8 - 2EF (COM 4) 338 - 33F 3E8 - 3EF (COM 3)
Primary Interrupt Select Default = 00h	70h (RW)	Bits[3:0] selects which interrupt level is used for the primary Interrupt. 00= no interrupt selected 01= IRQ1 02= IRQ2 03= IRQ3 04= IRQ4 05= IRQ5 06= IRQ6 07= IRQ7 08= IRQ8 09= IRQ9 0A= IRQ10 0B= IRQ11 0C= IRQ12 0D= IRQ13 0E= IRQ14 0F= IRQ15 Bits[7:4] Reserved Note: An Interrupt is activated by enabling this device (offset 30h), setting this register to a non-zero value and setting any combination of bits 0-4 in the corresponding UART IER and the occurrence of the corresponding UART event (i.e. Modem Status Change, Receiver Line Error Condition, Transmit Data Request, Receiver Data Available or Receiver Time Out) and setting the OUT2 bit in the MCR.
Reserved Default = 04h	74h	Bit 7:0 - Reserved
Reserved Default = 04h	75h	Bit 7:0 - Reserved
Reserved Default = 00h	F0h	Bit 7:0 - Reserved

Table 1080. Logical Device 6 (Watch Dog Timer)

Logical Device Register	Address	Description

Enable
Default = 00h
	30h
	(RW)
		Bits[7:1] Reserved, set to zero.
		Bit[0]
		1 =enable the logical device currently selected through the Logical Device # register.
		0 =Logical device currently selected is inactive

I/O Base Address Default = 00h	60h (RW) 61h (Bits 7:5 RW Bits 4:0 RO)	Registers 60h (MSB) and 61h (LSB) set the base address for the device. Note: Decode is on 32 Byte boundaries. WDT Base Address is generated by using the LPC Generic Decode Range 1 register (LG1), see D31:F0:84h for more details. This Base Address must be within the 128 bytes of LG1 Base register. Also the last byte accessed by the WDT must not exceed the LG1 Base Address +128 bytes.
Primary Interrupt Select Default = 00h	70h (RW)	Bits[3:0] selects which interrupt level is used for the primary Interrupt. 00= no interrupt selected 01= IRQ1 02= IRQ2 03= IRQ3 04= IRQ4 05= IRQ5 06= IRQ6 07= IRQ7 08= IRQ8 09= IRQ9 0A= IRQ10 0B= IRQ11 0C= IRQ12 0D= IRQ13 0E= IRQ14 0F= IRQ15 Bits[7:4] Reserved ***Note:*** An Interrupt is activated by enabling this device (offset 30h), setting this register to a non-zero value and when the first stage has been allowed to reach zero. An Interrupt is not generated if WDT_TOUT_CNF is set to change output after every timeout (See WDT Lock Register).

§ §

June 2007
Order Number: 313458-004US

32.0 Electrical Characteristics

32.1 Absolute Maximum Ratings

Table 1081 lists the maximum environmental stress ratings. Functional operation at the absolute maximum and minimum is neither implied nor guaranteed. Functional operating parameters are listed in the AC and DC tables.

Warning: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operating beyond the operating conditions is not recommended and extended exposure beyond operating conditions may affect reliability.

Table 1081. Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
$T_{STORAGE}$	Storage Temperature	–40.0	85.0	°C
V_{CC1_5}	Core Supply Voltage with respect to V_{SS}	–0.50	1.65	V
V_{TT}	Supply Voltage input with respect to V_{SS}	–0.50	1.1	V
V_{CC_DDR}	DDR2 Buffer Supply Voltage	–0.50	2.80	V

32.2 Power Characteristics

Table 1082. Operating Condition Power Supply Rails (Sheet 1 of 2)

Symbol	Parameter	Minimum		Nominal	Maximum		Unit	Notes
		AC	DC		AC	DC		
Vtt	Host AGTL+ Termination Voltage		1.019	1.05		1.082	V	1
Itt	Host AGTL+ Termination Current				785.0		mA	6
Vcc_CPU	Processor Interface Signals voltage		1.019	1.05		1.082	V	1
Vcc1_5	1.5 V Supply Voltage for the core well	1.425	1.455	1.5	1.575	1.545	V	1
Icc1_5	1.5V Supply Current on Vcc1_5				5.831		A	4, 6
Vcc_CORE_PLL	1.5 V Analog Voltage		1.455	1.5		1.545	V	2
Vcc_DDR	DDR2 Supply Voltage	1.674	1.71	1.8	1.926	1.89	V	1
Idd_DDR	DDR2 Supply Current				865		mA	3, 6
Vcc_DDR_PLL	DDR2 Analog Voltage		1.455V	1.5		1.545	V	2
Vcc_PEA_PLL	PCI Express Port "A" Analog voltage		1.455V	1.5		1.545	V	2, 6

Notes:

1. The DC min/max window describes the power supply behavior required at frequencies below 5 MHz. The AC min/max window describes the power supply behavior required at frequencies from 5 MHz to 2 GHz.
2. The analog voltage is intended to be a filtered copy of the 1.5 V core supply voltage. Refer to the Platform Design Guide for the reference implementation of the filters.
3. Idd_DDR x Vcc_DDR does not equal power dissipated from the DDR2 rail. Most of the current supplied by the Idd rail at max current draw is sourced out the memory signal pins at some voltage above ground.
4. Total current drawn off the 1.5V rail is separated into current drawn by the Vcc1_5 balls and the VCC_PEA_PLL and VCC_PEB_PLL balls as shown in the component ballout in the Intel® 3100 Chipset External Design Specification (EDS).
5. The VCC_PEA_PLL and VCC_PEB_PLL balls should be inductively isolated from the Vcc1_5 balls to avoid noise coupling from the core onto the VCC_PEA_PLL and VCC_PEB_PLL rails, which is particularly sensitive to AC noise. Refer to the Intel® 3100 Chipset Platform Design Guide for suggestions on how to achieve this isolation.
6. Guaranteed by design.
7. IccRTC data is taken with VccRTC at 3.0V while the system is in the G3 state at room temperature. Only the G3 state for this power well is shown to provide an estimated battery life.

Table 1082. Operating Condition Power Supply Rails (Sheet 2 of 2)

Symbol	Parameter	Minimum		Nominal	Maximum		Unit	Notes
		AC	DC		AC	DC		
Vcc_PEB_PLL	PCI Express Port "B" Analog voltage		1.455V	1.5		1.545	V	2, 6
Icc_PEA_PLL	PCI Express Port "A" Supply Current				4		mA	4, 5, 6
Icc_PEB_PLL	PCI Express Port "B" Supply Current				4		mA	4, 5, 6
Vcc_PEA_BG	PCI-Express Port "A" Analog Bandgap Voltage		2.425	2.5		2.575	V	6
Vcc_PEB_BG	PCI-Express Port "B" Analog Bandgap Voltage		3.135	3.3		3.465	V	6
Vcc_USB_PLL	USB Analog Voltage		1.455	1.5		1.545	V	
Vcc_USB_BG	USB Band Gap Voltage		3.135	3.3		3.465	V	
Vcc_SATA_PLL	SATA Analog Voltage		1.455	1.5		1.545	V	
Vcc_SATA_BG	SATA Band Gap Voltage		3.135	3.3		3.465	V	
Vcc3_3	PCI bus and Internal Logic Voltage	3.135	3.201	3.3	3.465	3.399	V	
Icc3_3	PCI and other I/O current				190		mA	
Vcc2_5	Internal Logic Voltage		2.375	2.5		2.625	V	
VccRTC	Battery/RTC Well Voltage		2.0	3.3		3.465	V	
IccRTC	RTC Leakage Current			6			µA	8
VccSus1_5	1.5 V Resume Well Voltage		1.455	1.5		1.545	V	
VccSus3_3	3.3 V Resume Well Voltage		3.135	3.3		3.465	V	
V5REF_Sus	Suspend Well Reference Voltage		4.75	5.0		5.25	V	
V5REF	Core Well Reference Voltage		4.75	5.0		5.25	V	

Notes:

1. The DC min/max window describes the power supply behavior required at frequencies below 5 MHz. The AC min/max window describes the power supply behavior required at frequencies from 5 MHz to 2 GHz.
2. The analog voltage is intended to be a filtered copy of the 1.5 V core supply voltage. Refer to the Platform Design Guide for the reference implementation of the filters.
3. Idd_DDR x Vcc_DDR does not equal power dissipated from the DDR2 rail. Most of the current supplied by the Idd rail at max current draw is sourced out the memory signal pins at some voltage above ground.
4. Total current drawn off the 1.5V rail is separated into current drawn by the Vcc1_5 balls and the VCC_PEA_PLL and VCC_PEB_PLL balls as shown in the component ballout in the Intel® 3100 Chipset External Design Specification (EDS).
5. The VCC_PEA_PLL and VCC_PEB_PLL balls should be inductively isolated from the Vcc1_5 balls to avoid noise coupling from the core onto the VCC_PEA_PLL and VCC_PEB_PLL rails, which is particularly sensitive to AC noise. Refer to the Intel® 3100 Chipset Platform Design Guide for suggestions on how to achieve this isolation.
6. Guaranteed by design.
7. IccRTC data is taken with VccRTC at 3.0V while the system is in the G3 state at room temperature. Only the G3 state for this power well is shown to provide an estimated battery life.

32.3 I/O Interface Signal Groupings

Signals are arranged in functional groups according to their associated interface.

The "#" symbol at the end of the signal name indicates that the active, or asserted state occurs when the signal is at a low voltage level. When "#" is not present, the signal is asserted when at the high voltage level.

The following notations are used to describe the signal type:

I	Input pin
O	Output pin
I/O	Bidirectional Input/Output pin
AGTL+	Open drain AGTL+ interface signal. Refer to the AGTL+ I/O Specification for complete details. The Intel® 3100 Chipset integrates AGTL+ termination resistors.
HCSL	The processor and the Intel® 3100 Chipset's reference clocks are HCSL, low-voltage differential CMOS technology.
CMOS	The CMOS buffered interface signals are handled by generic CMOS transceiver designs, operating on a full swing from rail to rail, with nominal threshold centered between them. The Intel® 3100 Chipset implements different CMOS voltage types which are listed in Table 1083.
SCHMITT	CMOS buffers with integrated Schmitt trigger circuit to compensate for slow edges. The IMCH implements different SCHMITT voltage types which are listed in Table 1083.
SSTL-2	The SSTL-2 technology is used for the source-synchronous DDR2 interface signals and are similar to standard CMOS pins.
DIFF	PCI Express* differential high speed serial signals.
OD	Open drain output. The Intel® 3100 Chipset can drive a logic 0 on the signal, but relies on an external pull-up resistor to attain a logic 1 level. The voltage associated with logic 1 is specified for each open drain type signal. The platform designer must ensure that these signals are not pulled to an unsupported voltage level.
Analog	References a signal type which does not follow standard digital logic levels. These signals include impedance compensation, power, ground, and other special characteristic signals.

Table 1083. CMOS and SCHMITT Signals Types

Type Name	Signal Technology	Voltage
CMOS1_05	CMOS	1.05
CMOS1_5	CMOS	1.5
CMOS1_8	CMOS	1.8
CMOS3_3	CMOS	3.3
CMOS5_0	CMOS	5.0
SCHMITT1_5	SCHMITT	1.5
SCHMITT3_3	SCHMITT	3.3
SCHMITT5_0	SCHMITT	5.0

Table 1084. FSB Signal Group

Signal Group	Signal Type	Signals	Notes
(1)	AGTL+ I/O	AP[1:0]#, ADS#, BNR#, DBSY#, DP[3:0]#, DRDY#, HA[35:3]#, HADSTB[1:0]#, HD[63:0] #, HDSTBp[3:0]#, HDSTBn[3:0]#, HIT#, HITM#, HREQ[4:0]#, BREQ[0]#, DINV[3:0]#, MCERR#	Table 1100
(2)	AGTL+ Output	BPRI#, CPURST#, DEFER#, HTRDY#, RS[2:0]#, RSP#	Table 1100
(3)	AGTL+ Input	HLOCK#, BINIT#, BREQ[1]#	Table 1100
(4)	Host Reference Voltage Input (Analog)	HDVREF[1:0], HACVREF	Table 1100
(5)	HCSL Clock Inputs	HCLKINn, HCLKINp	Table 1100
(6)	Schmitt1_5 Input	CPU_SEL[2:0]	Table 1100

Table 1085. Processor Side-Band Signal Group

Signal Group	Signal Type	Signals	Notes
(7)	CMOS1_5 output	A20M#, CPUSLP#, IGNNE#, INIT#, INTR, NMI, SMI#, STPCLK#	Table 1107
(8)	CMOS3_3 output	INIT3_3V#	Table 1107
(9)	OD output	CPUPWRGD/GPO[49][1]	Table 1107
(10)	CMOS1_5 input	FERR#	Table 1107
(11)	CMOS3_3 input	RCIN#, A20GATE	Table 1107

Note:
1. Signal has an open drain driver, and the V_{OH} specification does not apply. This signal must have external pull up resistor to 1.05.

Table 1086. DDR2 Signal Group

Signal Group	Signal Type	Signals	Notes
(12)	SSTL-2 I/O	DDR_DQ[63:0], DDR_CB[7:0], DDR_DQSp[17:0]DDR_DQSn[17:0]	Table 1101
(13)	Output	DDR_CMDCLKp[3:0], DDR_CMDCLKn[3:0], DDR_MA[14:0], DDR_BA[2:0], DDR_RAS#, DDR_CAS#, DDR_WE#, DDR_CS[7:0]#, DDR_CKE[3:0]	Table 1101
(14)	Memory Reference Voltage (Analog)	DDR_VREF	Table 1101

Table 1087. PCI Express* Signal Group

Signal Group	Signal Type	Signals	Notes
(15)	PCI-Express Input	PEA0_Rp[7:0], PEA0_Rn[7:0], PEB0_Rp[3:0], PEB0_Rn[3:0]	Table 1102
(16)	PCI-Express Output	PEA0_Tp[7:0], PEA0_Tn[7:0], PEB0_Tp[3:0], PEB0_Tn[3:0]	Table 1102
(17)	HCSL Clock input Pair	PEA_CLKp, PEA_CLKn, PEB_CLKp, PEB_CLKn	Table 1103

Table 1088. SMBus Signal Groups

Signal Group	Signal Type	Signals	Notes
(18)	CMOS3_3 Input	SMBALERT#/GPI[11]	Table 1104
(19)[1]	I/O	SMBDATA, SMBCLK, SMLINK[1:0]	Table 1104
(20)[1]	I/O	SMBSCL, SMBSDA	Table 1104

Note:
1. Signals have an open drain driver, and the V_{OH} spec does not apply. This signal must have external pull up resistor.

Table 1089. System Management and Power State Signal Groups

Signal Group	Signal Type	Signals	Notes
(21)	CMOS3_3 Input	THRMTRIP#	Table 1108
(22)	CMOS3_3 Output	SLP_S3#, SLP_S4#, SLP_S5#, SUS_STAT#/LPCPD#, SUSCLK, PLTRST#	Table 1108
(23)	CMOS3_3 Input	SYS_RESET#, PWRBTN#, RI#, WAKE#	Table 1108
(24)	CMOS3_3 Input	PWROK, RSMRST#, INTRUDER#	Table 1108
(25)	CMOS3_3 Input	VRMPWRGD/VGATE, THRM#	Table 1108
(26)	Schmitt3_3 Input	PWRGD, RSTIN#, PE_HPINITR#	Table 1108

Table 1090. Miscellaneous Signals and Clocks

Signal Group	Signal Type	Signals	Notes
(27)	CMOS5_0 Input	CLK14, CLK48	Table 1106
(28)	CMOS3_3 Output	SPKR	Table 1106
(29)	CMOS3_3 Input	INTVRMEN	Table 1106
(30)	CMOS3_3 Input	WL_PU1, WL_PU0	Table 1106
(31)	CMOS3_3 Output	PEB_RPC[1:0]	Table 1106

Table 1091. LPC/FWH Signal Group

Signal Group	Signal Type	Signals	Notes
(32)	CMOS3_3 Input	LDRQ[0]#, LDRQ[1]#/GPI[41]	Table 1109
(33)	CMOS3_3 Output	LFRAME#/FWH[4]	Table 1109
(34)	CMOS3_3 I/O	LAD[3:0]/FWH[3:0]	Table 1109

Table 1092. USB Signal Group

Signal Group	Signal Type	Signals	Notes
(35) (5 V Tolerant)	I/O	USBp[3:0], USBn[3:0] (Low-speed and Full-speed) or (in High-speed Mode)	Table 1110
(36)	Input	OC[3:0]#	Table 1110

Table 1093. Serial ATA Signal Group

Signal Group	Signal Type	Signals	Notes
(37)	Input	SATA_CLKn, SATA_CLKp	Table 1111
(38)	Input	SATA_RXp[5:0], SATA_RXn[5:0]	Table 1111
(39)	Output	SATA_TXp[5:0], SATA_TXn[5:0]	Table 1111
(40)	CMOS3_3 Input	SATA[5:0]_GP/GPI[13,12,31:29,26]	Table 1111
(41)[1]	OD Output	SATA_LED#	Table 1111

Note:
1. Signal has an open drain driver, and the V_{OH} spec does not apply. This signal must have external pull up resistor.

Table 1094. Interrupt Signal Group

Signal Group	Signal Type	Signals	Notes
(42)	CMOS3_3 I/O	SERIRQ	Table 1112
(43)	OD I/O	PIRQ[D:A]#, PIRQ[H:E]#/GPI[5:2]	1, Table 1112

Note:
1. These signals have an open drain driver, and the V_{OH} spec does not apply. These signals must have an external pull up resistor.

Table 1095. Watch Dog Timer and Real Time Clock Signal Group

Signal Group	Signal Type	Signals	Notes
(44)	CMOS3_3 Clock Input	RTCX[2:1]	Table 1113
(45)	CMOS3_3 Input	RTEST#	Table 1113
(46)	CMOS3_3 Output	WDT_TOUT#	Table 1113

Table 1096. General Purpose I/O Signal Group

Signal Group	Signal Type	Signals	Notes
(47)	CMOS5_0 Input	GPI[0], GPI[1], GPI[40]	Table 1114
(See Table 1094, Signal Group #43)		GPI[5:2]/PIRQ[H:E]#	Table 1094
(See Table 1088, Signal Group #18)		GPI[11]/SMBALERT#	Table 1088
(48)	CMOS3_3 Output	GPO[16], GPO[17], GPO[48]	Table 1114
(49)	CMOS3_3 Output	GPO[19:18], GPO[20]/SCLK[1], GPO[23]/SDATAOUT0[1], GPO[21]/SLOAD[1]	Table 1114
(50)	CMOS3_3 Input	GPI[8]	Table 1114
(See Table 1093, Signal Group #40)		GPI[13,12,31:29,26]/SATA[5:0]_GP	Table 1093
(51)	CMOS3_3 Input	GPI[15, 14, 10, 9, 7], GPI[6]/NC	Table 1114
(See Table 1091, Signal Group #32)		GPI[41]/LDRQ[1]#	Table 1091
(See Table 1085, Signal Group #9)		GPO[49]/CPUPWRGD	Table 1085
(52)	CMOS3_3 I/O	GPIO[28, 27, 25, 24]	Table 1114
(53)	CMOS3_3 I/O	GPIO[34, 33]	Table 1114
(54)	CMOS3_3 I/O	GPIO[32]/SDATAOUT1[1]	Table 1114

Note:

1. Signal has an open drain driver, and the V_{OH} spec does not apply. The signal must have an external pull up resistor.

Table 1097. UART Signal Group

Signal Group	Signal Type	Signals	Notes
(55)	CMOS3_3 Input	SIU_CTS[2:1]#, SIU_DCD[2:1]#, SIU_DSR[2:1]#, SIU_RI[2:1]#, SIU_RXD[2:1], UART_CLK	Table 1115
(56)	CMOS3_3 Output	SIU_TXD[2:1], SIU_DTR[2]#, SIU_RTS[2]#	Table 1115
(57)	CMOS3_3 I/O	SIU_DTR[1]#, SIU_RTS[1]#	Table 1115

Table 1098. PCI32/33 Signal Group

Signal Group	Signal Type	Signals	Notes
(58)	CMOS3_3 I/O	AD[31:0], C/BE[3:0]#, DEVSEL#, FRAME#, IRDY#, PAR, PERR#, PLOCK#, SERR#[1], STOP#, TRDY#	Table 1116
(59)	CMOS3_3 Input	REQ[1:0]#	Table 1116
(60)	CMOS3_3 Output	PCIRST#, GNT[1:0]#	Table 1116
(61)	CMOS3_3 Input	PCICLK	Table 1116
(62)	CMOS3_3 I/O	PME#[1]	Table 1116

Note:
1. Signal has an open drain driver, the V_{OH} spec does not apply. The signal must have an external pull up resistor.

Table 1099. TAP Signal Group

Signal Group	Signal Type	Signals	Notes
(63)	Schmitt1_5 Input	TRST#, TMS, TDI, TCK	Table 1105
(64)	OD Output	TDO	Table 1105
(65)	Schmitt1_5 Input	TEST#	Table 1105
(66)	CMOS1_5 I/O	Debug[7:0]	Table 1105

32.4 DC Characteristics

Table 1100. FSB Interface DC Characteristics (Sheet 1 of 3)

Signal Group	Symbol	Parameter	Min	Nom	Max	Unit	Notes
(1), (3)	V_{IL_H}	Host AGTL+ Input Low Voltage			GTLREF –0.10 x VTT	V	1
	V_{IH_H}	Host AGTL+ Input High Voltage	GTLREF + 0.10 x VTT			V	1
	I_{L_H}	Host AGTL+ Input Leakage Current			10	µA	
	C_{PAD}	Host AGTL+ Input Capacitance	1		2.5	pF	7

Note:
1. GTLREF is either HACVREF or HDVREF, depending on whether the input is an address, control, or a data signal
2. Crossing voltage is defined as the instantaneous voltage when the rising edge of HCLKINp is equal to the falling edge of HCLKINn.
3. VHavg is the statistical average of the VH measured by the oscilloscope.
4. Threshold Region is defined as a region entered around the crossing point voltage in which the differential receiver switches. It includes input threshold hysteresis.
5. The crossing point must meet the absolute and relative crossing point specifications simultaneously.
6. VHavg can be measured directly using “Vtop” on Agilent* scopes and “High” on Tektronix* scopes.
7. Guaranteed by design.

Table 1100. FSB Interface DC Characteristics (Sheet 2 of 3)

Signal Group	Symbol	Parameter	Min	Nom	Max	Unit	Notes
(1), (2)	V_{OL_H}	Host AGTL+ Output Low Voltage		0.4		V	
	V_{OH_H}	Host AGTL+ Output High Voltage	VTT – 0.1	VTT		V	
	I_{OL_H}	Host AGTL+ Output Low Current		17		mA	
	I_{Oh_H}	Host AGTL+ Output High Current				mA	
(4)	HACVREF	Host Address and Common clock Reference Voltage	0.98 x 1.05	0.63 x VTT	1.02 x 1.05	V	
	HDVREF	Host Data Reference Voltage	0.98 x 1.05	0.63 x VTT	1.02 x 1.05	V	
	R_{TT}	Host Termination Resistance	45	50	55	Ω	
(5)	V_{IL}	Input Low Voltage	–0.150	0	0.150	V	2
	V_{IH}	Input High Voltage	0.660	0.700	0.850	V	2
	$V_{CROSS(abs)}$	Absolute Crossing Point	0.250		0.550	V	2, 2, 5
	$V_{CROSS(rel)}$	Relative Crossing Point	0.250 + 0.5 x (V_{Havg} – 0.710)		0.550 + 0.5 x (V_{Havg} – 0.710)	V	2, 3, 5, 6, 7
	ΔV_{CROSS}	Range of Crossing Points			0.140	V	2
	V_{OS}	Overshoot			V_{IH}+ 0.300	V	2, 7
	V_{US}	Undershoot	–0.300			V	2, 7
	V_{RBM}	Ringback Margin	0.200			V	2, 7
	V_{TR}	Threshold Region	V_{CROSS} – 0.100		V_{CROSS} + 0.100	V	2, 4, 7
	C_{CLK}	Pin Capactiance	0.8 x Nominal	3.6	1.2 x Nominal	pF	7
(6)	$V_{IL_CPU_SEL}$	CPU_SEL[2:0] Input Low Voltage	-0.3		0.5	V	
	$V_{IH_CPU_SEL}$	CPU_SEL[2:0] Input High Voltage	1.0		VCC1_5 + 0.3	V	
	$I_{L_CPU_SEL}$	CPU_SEL[2:0] Input Leakage Current			50	µA	

Signal Group	Signal Type	Signals
(1)	AGTL+ I/O	AP[1:0]#, ADS#, BNR#, DBSY#, DP[3:0]#, DRDY#, HA[35:3]#, HADSTB[1:0] #, HD[63:0] #, HDSTBp[3:0]#, HDSTBn[3:0]#, HIT#, HITM#, HREQ[4:0]#, BREQ[0]#, DINV[3:0]#, MCERR#
(2)	AGTL+ Output	BPRI#, CPURST#, DEFER#, HTRDY#, RS[2:0]#, RSP#
(3)	AGTL+ Input	HLOCK#, BINIT#, BREQ[1]#

Note:

1. GTLREF is either HACVREF or HDVREF, depending on whether the input is an address, control, or a data signal
2. Crossing voltage is defined as the instantaneous voltage when the rising edge of HCLKINp is equal to the falling edge of HCLKINn.
3. VHavg is the statistical average of the VH measured by the oscilloscope.
4. Threshold Region is defined as a region entered around the crossing point voltage in which the differential receiver switches. It includes input threshold hysteresis.
5. The crossing point must meet the absolute and relative crossing point specifications simultaneously.
6. VHavg can be measured directly using “Vtop” on Agilent* scopes and “High” on Tektronix* scopes.
7. Guaranteed by design.

Table 1100. FSB Interface DC Characteristics (Sheet 3 of 3)

Signal Group	Symbol	Parameter	Min	Nom	Max	Unit	Notes
(4)	Reference Voltage (Analog)	HDVREF[1:0], HACVREF					
(5)	HCSL Clock Input	HCLKINn, HCLKINp					
(6)	Schmitt1_5 Input	CPU_SEL[2:0]					

Note:
1. GTLREF is either HACVREF or HDVREF, depending on whether the input is an address, control, or a data signal
2. Crossing voltage is defined as the instantaneous voltage when the rising edge of HCLKINp is equal to the falling edge of HCLKINn.
3. VHavg is the statistical average of the VH measured by the oscilloscope.
4. Threshold Region is defined as a region entered around the crossing point voltage in which the differential receiver switches. It includes input threshold hysteresis.
5. The crossing point must meet the absolute and relative crossing point specifications simultaneously.
6. VHavg can be measured directly using “Vtop” on Agilent* scopes and “High” on Tektronix* scopes.
7. Guaranteed by design.

Table 1101. DDR2 Interface (Sheet 1 of 2)DC Characteristics

Signal Group	Symbol	Parameter	Min	Nom	Max	Unit	Notes
(12)	$V_{IL\ (DC)}$	Input Low Voltage (DC)			DDR_VREF – 0.075	V	2
	$V_{IH\ (DC)}$	Input High Voltage (DC)	DDR_VREF + 0.075			V	2
	$V_{IL\ (AC)}$	Input Low Voltage (AC)			DDR_VREF – 0.175	V	2
	$V_{IH\ (AC)}$	Input High Voltage (AC)	DDR_VREF + 0.175			V	2
	I_{Leak}	Input Leakage Current			5.0	µA	
(12), (13)	I_{OH}	Output High Current			20.7	mA	3
	I_{OL}	Output Low Current			18	mA	3
	V_{OL}	Output Low Voltage	0.0		0.414	V	3
	V_{OH}	Output High Voltage	1.3			V	3
	C_{Pin}	Pin Capacitance	2.5		3.75	pF	4
(14)	DDRVREF	DDR2 Reference Voltage	0.49 x Vcc_DDR	0.5 x Vcc_DDR	0.51 x Vcc_DDR	V	
Signal Group	**Signal Type**	**Signals**					

Notes:
1. Refer to the JESD-79-2 and SSTL-18 Specification for further details.
2. These input voltages apply only when the signals are inputs to Intel® 3100 Chipset. When the signals are inputs to the SDRAM, the SDRAM input voltage specifications apply.
3. DDR2 DC parameters are specified in a 43 ohm test load to Vdd/2.
4. Guaranteed by design.

Table 1101. DDR2 Interface (Sheet 2 of 2)DC Characteristics

Signal Group	Symbol	Parameter	Min	Nom	Max	Unit	Notes
(12)	SSTL-2 I/O	DDR_DQ[63:0], DDR_CB[7:0], DDR_DQSp[17:0], DDR_DQSn[17:0]					
(13)	Output	DDR_CMDCLKp[3:0], DDR_CMDCLKn[3:0], DDR_MA[14:0], DDR_BA[2:0], DDR_RAS#, DDR_CAS#, DDR_WE#, DDR_CS[7:0]#, DDR_CKE[7:0]					
(14)	Reference Voltage (Analog)	DDR_VREF					

Notes:
1. Refer to the JESD-79-2 and SSTL-18 Specification for further details.
2. These input voltages apply only when the signals are inputs to Intel® 3100 Chipset. When the signals are inputs to the SDRAM, the SDRAM input voltage specifications apply.
3. DDR2 DC parameters are specified in a 43 ohm test load to Vdd/2.
4. Guaranteed by design.

Table 1102. PCI Express* Differential Transmitter (TX) and Receiver (RX) DC Specifications

Signal Group	Symbol	Parameter	Min	Nom	Max	Unit	Notes
(15)	$Z_{RX-DIFF-DC}$	RX DC Differential Input Impedance	80	100	120	Ω	
	Z_{RX-DC}	RX DC Input Common Mode Impedance	40	50	60	Ω	, 5
	$Z_{RX-HIGH-IMP-DC}$	RX Powered Down DC Input Common Mode Impedance	200k			Ω	6
	$L_{RX-SKEW}$	RX Total Skew			20	ns	
	V_{I8}	RX Input Voltage	175		1200	mVdiffp-p	4
(16)	$V_{TX-CM-DC-ACTIVE-IDLE-DELTA}$	TX Absolute Delta of DC Common Mode Voltage During L0 and Electrical Idle	0		100	mV	
	$V_{TX-CM-DC-LINE-DELTA}$	TX Absolute Delta of DC Common Mode Voltage between D+ and D–	0		25	mV	
	$V_{TX-DC-CM}$	TX DC Common Mode Voltage	0		3.6	V	1
	$I_{TX-SHORT}$	TX Short Circuit Current Limit			90	mA	3
	$Z_{TX-DIFF-DC}$	TX DC Differential Impedance	80	100	120	Ω	
	Z_{TX-DC}	TX DC impedance	40			Ω	
	$L_{TX-SKEW}$	TX Lane-to-Lane Output Skew			500 ps + 2UI		2
	C_{TX}	TX AC Coupling Capacitor	75		200	nF	7
	V_{O7}	TX Output Voltage	800		1200	mVdiffp-p	4

Signal Group	Signal Type	Signals
(15)	RX Input	PEA0_Rp[7:0], PEA0_Rn[7:0], PEB0_Rp[1:0], PEB0_Rn[1:0]
(16)	TX Output	PEA0_Tp[7:0], PEA0_Tn[7:0], PEB0_Tp[1:0], PEB0_Tn[1:0]

Note:
1. The allowed DC Common Mode voltage under any conditions. Refer to Section 4.3.1.8 in the *PCI Express* Specification* for further details.
2. Static skew between any two transmitter lanes within a single link
3. The allowed current when any output is shorted to ground.
4. PCI Express mVdiff p-p = PEx_Xp[x] - PEx_Xn[x]
5. Impedance during all LTSSM states. When transitioning from a Fundamental Reset to Detect (the initial state of the LTSSM) there is a 5 ms transition time before Receiver termination values must be met on all un-configured Lanes of a Port.
6. The RX DC Common Mode Impedance that exists when no power is present or Fundamental Reset is asserted. This helps ensure that the Receiver Detect circuit will not falsely assume a Receiver is powered on when it is not. This term must be measured at 300 mV above the RX ground.
7. Guaranteed by design.

Table 1103. PCI Express* Clock DC Characteristics

Sig Group	Symbol	Parameter	Min	Nom	Max	Unit	Notes
(17)	V_{IL}	Input Low Voltage	–0.150	0	0.150	V	1
	V_{IH}	Input High Voltage	0.660	0.710	0.850	V	1
	$V_{CROSS(abs)}$	Absolute Crossing Point	0.250		0.550	V	1, 1, 7
	$V_{CROSS(rel)}$	Relative Crossing Point	0.250 + 0.5 x (V_{Havg} – 0.710)		0.550 + 0.5 x (V_{Havg} – 0.710)	V	1, 2, 7, 8
	ΔV_{CROSS}	Range of Crossing Points			0.140	V	1
	V_{OS}	Overshoot			V_{IH}+ 0.300	V	1, 3
	V_{US}	Undershoot	–0.300			V	1, 4
	V_{RBM}	Ringback Margin	0.200			V	1, 5
	V_{TR}	Threshold Region	V_{CROSS} – 0.100		V_{CROSS} + 0.100	V	1, 6

Sig Group	Sig type	Signals
(17)	Differential Clocks	PEA_CLKp, PEA_CLKn, PEB_CLKn and PEB_CLKp (100 MHz Reference Clock)

Notes:

1. Crossing voltage is defined as the instantaneous voltage when the rising edge of HCLKINp is equal to the falling edge of HCLKINn.
2. VHavg is the statistical average of the VH measured by the oscilloscope.
3. Overshoot is defined as the absolute value of the maximum voltage.
4. Undershoot is defined as the value of the absolute minimum voltage.
5. Ringback Margin is defined as the absolute voltage difference between the maximum Rising Edge Ringback and the maximum Falling Edge Ringback.
6. Threshold Region is defined as a region entered around the crossing point voltage in which the differential receiver switches. It includes input threshold hysteresis.
7. The crossing point must meet the absolute and relative crossing point specifications simultaneously.
8. VHavg can be measured directly using “Vtop” on Agilent* scopes and “High” on Tektronix* scopes.

Table 1104. SMBus I/O DC Characteristics

Sig Group	Symbol	Parameter	Min	Max	Unit	I_{OL}/I_{OH}	Notes
(18) (19)	V_{ih}	Input high voltage	2.1	VccSus3_3 + 0.3	V		
	V_{il}	Input low voltage	-0.3	0.8	V		
(20)	V_{ih}	Input high voltage	2.1	Vcc3_3 + 0.3	V		
	V_{il}	Input low voltage	-0.3	0.8	V		
(19)[1], (20)[1]	I_{oh}	Output high current	-2 mA				
	V_{ol}	Output low voltage		0.4	V	4 mA	Ipullup = 4 mA
	I_{ol}	Output low current		4	mA		
	I_{leak}	Leakage current		10	µA		
	C_{pad}	Pad capacitance		10	pF		pad only, 2
Signal Group	**Type**	**Signals**					
(18)	CMOS3_3 Input	SMBALERT#/GPI[11]					
(19)[1]	I/O	SMBDATA, SMBCLK, SMLINK[1:0]					
(20)[1]	I/O	SMBSCL,SMBSDA					

Note:
1. Signals have an open drain driver therefore the V_{OH} spec does not apply. This signal must have an external pull up resistor.
2. Guaranteed by design.

Table 1105. TAP DC Characteristics

Sig Group	Symbol	Parameter	Min	Max	Unit	Notes
(63)	V_{ih}	Input high voltage	1.0		V	
	V_{il}	Input low voltage		0.5	V	
	I_{leak}	Leakage current		10	µA	
(64)	V_{ol}	Output low voltage		0.4	V	Rpullup = 50 Ω
(65), (66)	V_{ih}	Input high voltage	1.0	Vcc1_5 + 0.3	V	
	V_{il}	Input low voltage	-0.3	0.5	V	
	I_{leak}	Leakage current		50	µA	
(66)	V_{ol}	Output low voltage		0.1(Vcc1_5)		
	V_{oh}	Output High voltage	0.9(Vcc1_5)			
Sig Group	**Type**	**Signals**				
(63)	Schmitt1_5 Input	TRST#, TMS, TDI, TCK				
(64)	OD Output	TDO				
(65)	Schmitt1_5Input	TEST#				
(66)	CMOS1_5 I/O	Debug[7:0]				

Table 1106. Miscellaneous I/O DC Characteristics

Sig Group	Symbol	Parameter	Min	Max	Unit	I_{OL} / I_{OH}	Notes
(27)	V_{IL3}	Input Low Voltage	–0.5	0.8	V		2
	V_{IH3}	Input High Voltage	2.0	Vcc3_3 + 0.5	V		2
	I_{LI4}	Input Leakage Current	–100	+100	µA		
(28)	V_{OL5}	Output Low Voltage		0.4	V	6 mA	
	V_{OH5}	Output High Voltage	Vcc3_3 - 0.5		V	-2 mA	
(29)	V_{IL14}	Input Low Voltage	-0.5	0.8	V		
	V_{IH14}	Input High Voltage	2.0	VccRTC + 0.5	V		1
(30)	V_{IL_PU}	Input Low Voltage	-0.5	0.8	V		
	V_{IH_PU}	Input High Voltage	2.0	Vcc3_3 + 0.5	V		
(31)	V_{OL3}	Input Low Voltage		0.1(Vcc3_3)	V	6 mA	
	V_{OH3}	Input High Voltage	0.9(VCC3_3)		V	-0.5 mA	
Sig Group	**Type**	**Signals**					
(27)	CMOS5_0 Clocks	CLK14, CLK48					
(28)	CMOS3_3 Output	SPKR					
(29)	CMOS3_3 Input	INTVRMEN					
(30)	CMOS3_3 Input	WL_PU1, WL_PU0					
(31)	CMOS3_3 Output	PEB_RPC[1:0]					

Note:

1. V_{ccRTC} is the voltage applied to the V_{ccRTC} well. When the system is in a G3 state, this is generally supplied by the coin cell battery and can drop to 2 V, but for S5, this is generally VccSus3_3.
2. Signals are 5.0 V tolerant.

Table 1107. Processor Side-Band DC Characteristics

Sig Group	Symbol	Parameter	Min	Max	Unit	I_{OL} / I_{OH}	Notes
(7)	V_{OL2}	Output Low Voltage		0.255	V	3.0mA	
	V_{OH2}	Output High Voltage	Vcc_CPU -0.3		V	-0.3mA	
(8)	V_{OL5}	Output Low Voltage		0.4	V	6 mA	
	V_{OH5}	Output High Voltage	Vcc3_3 - 0.5		V	-2 mA	
(9)	V_{OL10}	Output Low Voltage		0.125	V	3 mA	2
	V_{OH10}	Output High Voltage	N/A		V		1
(10)	V_{IL7}	Input Low Voltage	–0.5	0.58(Vcc_CPU)	V		
	V_{IH7}	Input High Voltage	0.73(Vcc_CPU)	Vcc_CPU + 0.5	V		
(11)	V_{IL4}	Input Low Voltage	–0.5	0.3(Vcc3_3)	V		
	V_{IH4}	Input High Voltage	0.5(Vcc3_3)	Vcc3_3 + 0.5	V		

Signal Group	Signal Type	Signals
(7)	CMOS1_5 output	A20M#, CPUSLP#, IGNNE#, INIT#, INTR, NMI, SMI#, STPCLK#
(8)	CMOS3_3 output	INIT3_3V#
(9)	OD output	CPUPWRGD/GPO[49]
(10)	CMOS1_5 input	FERR#
(11)	CMOS3_3 input	RCIN#, A20GATE

Note:

1. The CPUPWRGD signal has an open drain driver and the V_{OH} spec does not apply. This signal must have external pull up resistor.
2. Listed I_{OL}/I_{OH} is typical. Maximum is 12 mA for short durations (<500 mS) and 9 mA for long durations.

Table 1108. System Management and Power State DC Characteristics

Signal Group	Symbol	Parameter	Min	Max	Unit	I_{OL} / I_{OH}	Notes
(21)	V_{IL7}	Input Low Voltage	–0.5	0.58(Vcc_CPU)	V		
	V_{IH7}	Input High Voltage	0.73(Vcc_CPU)	Vcc_CPU + 0.5	V		
(22)	V_{OL5}	Output Low Voltage		0.4	V	6 mA	
	V_{OH5}	Output High Voltage	Vcc3_3 - 0.5		V	-2 mA	
(23)	V_{IL13}	Input Low Voltage	–0.5	0.8	V		
	V_{IH13}	Input High Voltage	2.0	VccSus3_3 + 0.5	V		
(24)	V_{IL14}	Input Low Voltage	–0.5	0.8	V		
	V_{IH14}	Input High Voltage	2.0	VccRTC + 0.5	V		1
(25)	V_{IL3}	Input Low Voltage	–0.5	0.8	V		
	V_{IH3}	Input High Voltage	2.0	Vcc3_3 + 0.5	V		
(26)	V_{ih}	Input high voltage	2.1	Vcc3_3 + 0.3	V		
	V_{il}	Input low voltage	-0.3	0.8	V		
	I_{leak}	Leakage current		10	µA		
	C_{pad}	Pad capacitance		5	pF		pad only, 2
Signal Group	**Signal Type**	**Signals**					
(21)	CMOS3_3	THRMTRIP#					
(22)	CMOS3_3	SLP_S3#, SLP_S4#, SLP_S5#, SUS_STAT#/LPCPD#, SUSCLK, PLTRST#					
(23)	CMOS3_3	SYS_RESET#, PWRBTN#, RI#, WAKE#					
(24)	CMOS3_3	PWROK, RSMRST#, INTRUDER#					
(25)	CMOS3_3	VRMPWRGD/VGATE, THRM#					
(26)	Schmitt3_3 Input	PWRGD, RSTIN#, PE_HPINTR#					

Note:

1. V_{ccRTC} is the voltage applied to the V_{ccRTC} well. When the system is in a G3 state, this is generally supplied by the coin cell battery and can drop to 2 V, but for S5, this is generally VccSus3_3.
2. Guaranteed by design.

Table 1109. LPC/FWH DC Characteristics

Sig Group	Symbol	Parameter	Min	Max	Unit	I_{OL} / I_{OH}	Notes
(32), ()	V_{IL4}	Input Low Voltage	–0.5	0.3(Vcc3_3)	V		
	V_{IH4}	Input High Voltage	0.5(Vcc3_3)	Vcc3_3 + 0.5	V		
(33), ()	V_{OL9}	Output Low Voltage		0.1(Vcc3_3)	V	1.5 mA	
	V_{OH9}	Output High Voltage	0.9(Vcc3_3)		V	-0.5 mA	

Sig Group	Type	Signals
(32)	CMOS3_3 Input	LDRQ[0]#, LDRQ[1]#/GPI[41]
(33)	CMOS3_3 Output	LFRAME#/FWH[4]
()	CMOS3_3 I/O	LAD[3:0]/FWH[3:0]

Table 1110. USB DC Characteristics

Sig Group	Symbol	Parameter	Min	Max	Unit	I_{OL} / I_{OH}	Notes
(35)	V_{DI}	Input Differential Input Sensitivity	0.2		V		1, 2, 4, 5
	V_{CM}	Input Differential Common Mode Range	0.8	2.5	V		1, 2, 3, 5
	V_{SE}	Input Single-Ended Receiver Threshold	0.8	2.0	V		1, 2, 5
	V_{CRS}	Output Signal Crossover Voltage	1.3	2.0	V		2
	V_{HSSQ}	Input HS Squelch Detection Threshold	100	150	mV		1, 5
	V_{HSDSC}	Input HS Disconnect Detection Threshold	525	625	mV		1, 5
	V_{HSCM}	Input HS Data Signaling Common Mode Voltage Range	–50	500	mV		1, 5
	V_{HSOI}	Output HS Idle Level	–10.0	10.0	mV		1
	V_{HSOH}	Output HS Data Signaling High	360	440	mV		1
	V_{HSOL}	Output HS Data Signaling Low	–10.0	10.0	mV		1
	V_{CHIRPJ}	Output Chirp J Level	700	1100	mV		1
	V_{CHIRPK}	Output Chirp K Level	–900	–500	mV		1
	V_{OL6}	Output Low Voltage		0.4	V	5 mA	2
	V_{OH6}	Output High Voltage	Vcc3_3 – 0.5		V	-2 mA	2
(36)	V_{IL4}	Input Low Voltage	–0.5	0.3(Vcc3_3)	V		
	V_{IH4}	Input High Voltage	0.5(Vcc3_3)	Vcc3_3 + 0.5	V		

Sig Group	Type	Signals
(35)	I/O	USBp[3:0], USBn[3:0] (Low-speed and Full-speed) or (High-speed Mode)
(36)	Input	OC[3:0]#

Note:
1. Applies to High-speed USB 2.0
2. Applies to Low-speed and Full-speed
3. Includes V_{DI} range
4. V_{DI} = USBp[X] - USBn[X]
5. Signals are 5 V tolerant

Table 1111. Serial ATA DC Characteristics

Sig Group	Symbol	Parameter	Min	Max	Unit	I_{OL} / I_{OH}	Notes
(37)	V_{IL12}	Input Low Voltage	-0.150	0.150	V		
	V_{IH12}	Input High Voltage	0.660	0.850	V		
	$V_{cross(abs)}$	Absolute Crossing Point	0.250	0.550	V		
(38)	V_{IMIN10}	Minimum Input Voltage	325		mVdiffp-p		1
	V_{IMAX10}	Maximum Input Voltage		600	mVdiffp-p		1
(39)	V_{OMIN8}	Minimum Output Voltage	400		mVdiffp-p		2
	V_{OMAX8}	Maximum Output Voltage		600	mVdiffp-p		2
(40)	V_{IL3}	Input Low Voltage	–0.5	0.8	V		
	V_{IH3}	Input High Voltage	2.0	Vcc3_3 + 0.5	V		
(41)	V_{OL5}	Output Low Voltage		0.4	V	6 mA	3

Signal Group	Signal Type	Signals
(37)	Input	SATA_CLKn, SATA_CLKp
(38)	Input	SATA_RXp[5:0], SATA_RXn[5:0]
(39)	Output	SATA_TXp[5:0], SATA_TXn[5:0]
(40)	CMOS3_3 Input	SATA[5:0]_GP/GPI[13,12,31:29,26]
(41)	OD Output	SATA_LED#

Notes:

1. SATA Vdiff, tx ($V_{IMAX/MIN10}$ is measured at the SATA connector on the transmit side (generally, the motherboard connector)), where SATA mVdiff p-p = |SATA_RXp[x] - SATA_RXn[x]|
2. SATA Vdiff, tx ($V_{OMAX/MIN8}$ is measured at the SATA connector on the transmit side (generally, the motherboard connector)), where SATA mVdiff p-p = |SATA_TXp[x] - SATA_TXn[x]|
3. SATA_LED# has an open collector driver and the V_{OH} spec does not apply. This signal must have external pull up resistor.

Table 1112. Interrupt's DC Characteristics

Sig Group	Symbol	Parameter	Min	Max	Unit	I_{OL} / I_{OH}	Notes
(42)	V_{IL4}	Input Low Voltage	–0.5	0.3(Vcc3_3)	V		
	V_{IH4}	Input High Voltage	0.5(Vcc3_3)	Vcc3_3 + 0.5	V		
	V_{OL9}	Output Low Voltage		0.1(Vcc3_3)	V	1.5 mA	
	V_{OH9}	Output High Voltage	0.9(Vcc3_3)		V	-0.5 mA	
(43)	V_{IL1}	Input Low Voltage	–0.5	0.3(Vcc3_3)	V		2
	V_{IH1}	Input High Voltage	0.5(Vcc3_3)	V5REF + 0.5	V		1, 2

Sig Group	Type	Signals
(42)	CMOS3_3 I/O	SERIRQ
(43)	OD Input	PIRQ[D:A]#, PIRQ[H:E]#/GPI[5:2]

Note:
1. Signals have an open drain driver, and the V_{OH} spec does not apply. These signals must have an external pull up resistor.
2. Signals are 5.0 V tolerant.

Table 1113. Watch Dog Timer and Real Time Clock DC Characteristics

Sig Group	Symbol	Parameter	Min	Max	Unit	I_{OL} / I_{OH}	Notes
(44)	V_{IL9}	Input Low Voltage	–0.5	0.10	V		
	V_{IH9}	Input High Voltage	0.40	1.2	V		
(45)	V_{IL3}	Input Low Voltage	–0.5	0.8	V		
	V_{IH3}	Input High Voltage	2.0	Vcc3_3 + 0.5	V		
(46)	V_{OL6H}	Output Low Voltage		0.4	V	4 mA	
	V_{OH6H}	Output High Voltage	Vcc3_3 - 0.5		V	-2 mA	
(44)	C_L	XTAL1 typical value	6		pF		
	C_L	XTAL2 typical value	6		pF		

Sig Group	Type	Signals
(44)	Clock Input	RTCX[2:1]
(45)	CMOS Input	RTEST#
(46)	CMOS3_3 Output	WDT_TOUT#

Table 1114. General Purpose I/O DC Characteristics

Sig Group	Symbol	Parameter	Min	Max	Unit	I_{OL}/I_{OH}	Notes
(47)	V_{IL1}	Input Low Voltage	–0.5	0.3(Vcc3_3)	V		
	V_{IH1}	Input High Voltage	0.5(Vcc3_3)	V5REF + 0.5	V		
(48), (54)	V_{OL9}	Output Low Voltage		0.1(Vcc3_3)	V	1.5 mA	
	V_{OH9}	Output High Voltage	0.9(Vcc3_3)		V	-0.5 mA	
(49), (53), (52)	V_{OL5}	Output Low Voltage		0.4	V	6 mA	
	V_{OH5}	Output High Voltage	Vcc3_3 - 0.5		V	-2 mA	
(50), (53)	V_{IL3}	Input Low Voltage	–0.5	0.8	V		
	V_{IH3}	Input High Voltage	2.0	Vcc3_3 + 0.5	V		
(51), (54)	V_{IL4}	Input Low Voltage	–0.5	0.3(Vcc3_3)	V		
	V_{IH4}	Input High Voltage	0.5(Vcc3_3)	Vcc3_3 + 0.5	V		
(52)	V_{IL13}	Input Low Voltage	–0.5	0.8	V		
	V_{IH13}	Input High Voltage	2.0	VccSus3_3 + 0.5	V		

Sig Group	Type	Signals
(47)	CMOS5_0 Input	GPI[0], GPI[1], GPI[40]
(48)	CMOS3_3 Output	GPO[16, 17, 48]
(49)	CMOS3_3 Output	GPO[19:18], GPO[20]/SCLK[1], GPO[23]/SDATAOUT0[1], GPO[21]/SLOAD[1]
(50)	CMOS3_3 Input	GPI[8]
(51)	CMOS3_3 Input	GPI[15, 14, 10, 9, 7], GPI[6]/NC
(52)	CMOS3_3 I/O	GPIO[28, 27, 25, 24]
(53)	CMOS3_3 I/O	GPIO[34, 33]
(54)	CMOS3_3 I/O	GPIO[32]/SDATAOUT1[1]
The DC characteristics are defined by the muxed signal	CMOS Input	GPI[41]/LDRQ1# (See Table 1109, Signal Group #32)
	CMOS Output	GPO[49]CPUPWRGD (See Table 1107, Signal Group #9)
	CMOS Input	GPI[5:2]/PIRQ[H:E]# (See Table 1112, Signal Group #43)
	CMOS3_3 Input	GPI[11]/SMBALERT# (See Table 1104, Signal Group #18)
	CMOS3_3 Input	GPI[31:29, 26,13,12,]/SATA[5:0]_GP (See Table 1111, Signal Group #40)
	CMOS Output	GPO[49]/CPUPWRGD (See Table 1107, Signal Group #9)

Note:

1. Signal has an open drain driver, and the V_{OH} spec does not apply. The signal must have an external pull up resistor.

Table 1115. UART DC Characteristics

Sig Group	Symbol	Parameter	Min	Max	Unit	I_{OL}/I_{OH}	Notes
(55), (57)	V_{IL4}	Input Low Voltage	–0.5	0.3(Vcc3_3)	V		
	V_{IH4}	Input High Voltage	0.5(Vcc3_3)	Vcc3_3 + 0.5	V		
(56), (57)	V_{OL3}	Output Low Voltage		0.1(Vcc3_3)	V	6 mA	
	V_{OH3}	Output High Voltage	0.9(Vcc3_3)		V	-0.5 mA	

Sig Group	Type	Signals
(55)	CMOS3_3 Input	SIU_CTS[2:1]#, SIU_DCD[2:1]#, SIU_DSR[2:1]#, SIU_RI[2:1]#, SIU_RXD[2:1], UART_CLK
(56)	CMOS3_3 Output	SIU_TXD[2:1], SIU_DTR[2]#, SIU_RTS[2]#
(57)	CMOS3_3 I/O	SIU_DTR[1]#, SIU_RTS[1]#

Table 1116. PCI32/33 DC Characteristics (Sheet 1 of 2)

Sig Group	Symbol	Parameter	Min	Max	Unit	I_{OL}/I_{OH}	Notes
(58), (59)	V_{IL1}	Input Low Voltage	–0.5	0.3(Vcc3_3)	V		1
	V_{IH1}	Input High Voltage	0.5(Vcc3_3)	V5REF + 0.5	V		1
(58)	V_{OL4}	Output Low Voltage		0.55	V	6 mA	1
	V_{OH4}	Output High Voltage	0.9(Vcc3_3)		V	-0.5 mA	1
(60)	V_{OL9}	Output Low Voltage		0.1(Vcc3_3)	V	1.5 mA	
	V_{OH9}	Output High Voltage	0.9(Vcc3_3)		V	-0.5 mA	
(61), (62)	V_{IL4}	Input Low Voltage	–0.5	0.3(Vcc3_3)	V		
	V_{IH4}	Input High Voltage	0.5(Vcc3_3)	Vcc3_3 + 0.5	V		
	I_{LI4}	Input Leakage Current	–100	+100	µA		3
(62)	V_{OL11}	Output Low Voltage		0.4	V	1.5 mA	2
(58), (59), (60), (61)	I_{LI2}	PCI_3V Hi-Z State Data Line Leakage	–10	10	µA		(0 V < V_{IN} < 3.3V)
(58), (59)	I_{LI3}	PCI_5V Hi-Z State Data Line Leakage	–70	70	µA		1, Max V_{IN} = 2.7 V Min V_{IN} = 0.5 V

Sig Group	Type	Signals
(58)	CMOS3_3 I/O	AD[31:0], C/BE[3:0]#, DEVSEL#, FRAME#, IRDY#, PAR, PERR#, PLOCK#, SERR#[2], STOP#, TRDY#
(59)	CMOS3_3 Input	REQ[1:0]#

Notes:
1. Signals are 5.0 V tolerant.
2. Signal has an open drain driver, and the V_{OH} spec does not apply. The signal must have an external pull up resistor.
3. Specification only applies to PCICLK.

Table 1116. PCI32/33 DC Characteristics (Sheet 2 of 2)

(60)	CMOS3_3 Output	PCIRST#, GNT[1:0]#
(61)	CMOS3_3 Input	PCICLK
(62)	CMOS3_3 I/O	PME#

Notes:
1. Signals are 5.0 V tolerant.
2. Signal has an open drain driver, and the V_{OH} spec does not apply. The signal must have an external pull up resistor.
3. Specification only applies to PCICLK.

Table 1117. Other I/O DC Characteristics

Symbol	Parameter	Min	Max	Unit	Notes
C_{IN}	Input Capacitance — All Other		12	pF	F_C = 1 MHz, 1, 2
C_{OUT}	Output Capacitance		12	pF	F_C = 1 MHz, 1, 2
$C_{I/O}$	I/O Capacitance		12	pF	F_C = 1 MHz, 1, 2

Notes:
1. Unless stated otherwise these apply to all interfaces except FSB and DDR2 signals.
2. Guaranteed by design.